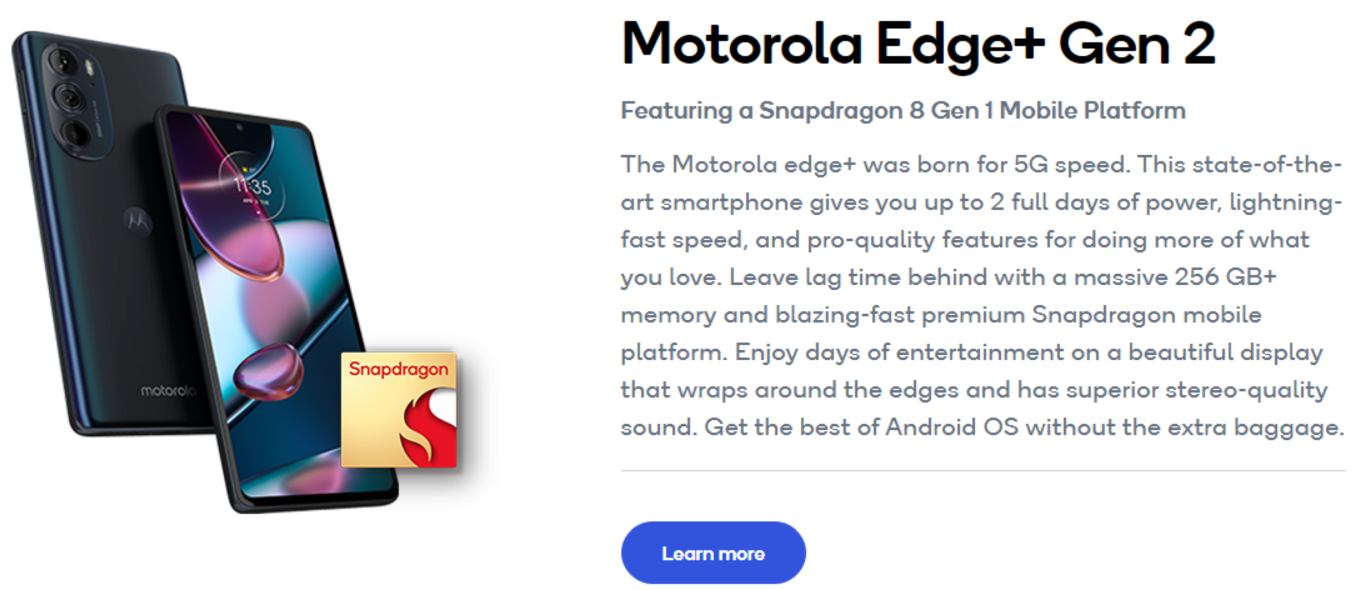
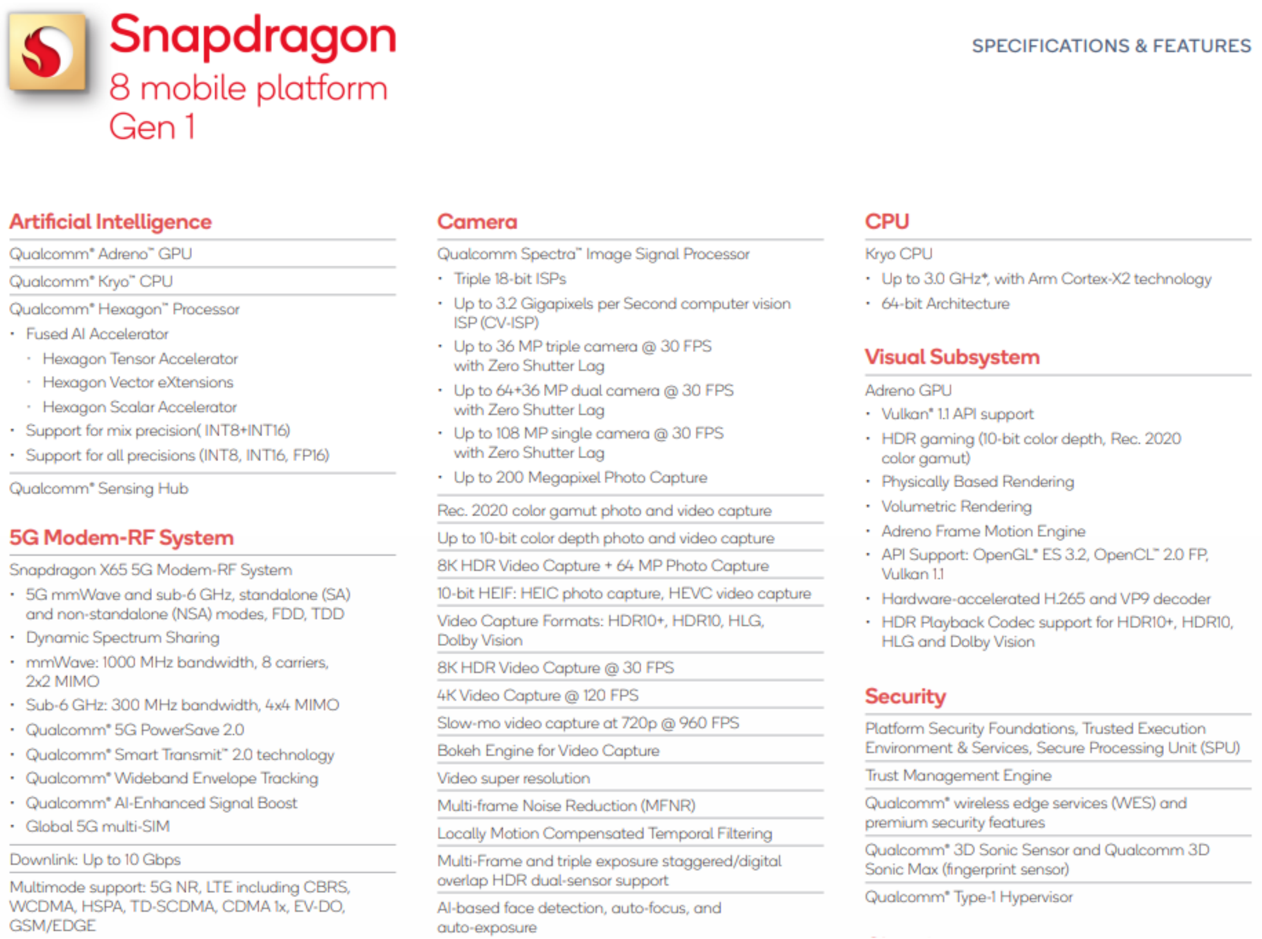
EXHIBIT 035

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
4. A method for exchanging messages in an integrated circuit comprising a plurality of modules, the messages between the plurality of modules being exchanged via a network	Without conceding that the preamble of claim 4 of the '9893 Patent is limiting, the Motorola Edge+ Gen 2 (hereinafter, the "Motorola product") performs a method for exchanging messages in an integrated circuit comprising a plurality of modules, the messages between the plurality of modules being exchanged via a network, either literally or under the doctrine of equivalents. The Motorola product includes an integrated circuit. For example, the Motorola product includes the Qualcomm Snapdragon 8 Gen 1 Mobile Platform system on chip (hereinafter, the "Snapdragon SoC").  Motorola Edge+ Gen 2 Featuring a Snapdragon 8 Gen 1 Mobile Platform The Motorola edge+ was born for 5G speed. This state-of-the-art smartphone gives you up to 2 full days of power, lightning-fast speed, and pro-quality features for doing more of what you love. Leave lag time behind with a massive 256 GB+ memory and blazing-fast premium Snapdragon mobile platform. Enjoy days of entertainment on a beautiful display that wraps around the edges and has superior stereo-quality sound. Get the best of Android OS without the extra baggage.

[1] The Motorola product is charted as a representative product made used, sold, offered for sale, and/or imported by Motorola. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	https://www.qualcomm.com/snapdragon/device-finder/motorola-edge--gen-2 The Snapdragon SoC comprises a plurality of modules, for example Qualcomm Adreno GPU; Qualcomm Kryo CPU; Qualcomm Hexagon Processor; and Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU): 

Snapdragon
8 mobile platform
Gen 1

SPECIFICATIONS & FEATURES

Artificial Intelligence

Qualcomm® Adreno™ GPU

Qualcomm® Kryo™ CPU

Qualcomm® Hexagon™ Processor
- Fused AI Accelerator
  - Hexagon Tensor Accelerator
  - Hexagon Vector eXtensions
  - Hexagon Scalar Accelerator
- Support for mix precision(INT8+INT16)
- Support for all precisions (INT8, INT16, FP16)

Qualcomm® Sensing Hub

5G Modem-RF System

Snapdragon X65 5G Modem-RF System
- 5G mmWave and sub-6 GHz, standalone (SA) and non-standalone (NSA) modes, FDD, TDD
- Dynamic Spectrum Sharing
- mmWave: 1000 MHz bandwidth, 8 carriers, 2x2 MIMO
- Sub-6 GHz: 300 MHz bandwidth, 4x4 MIMO
- Qualcomm® 5G PowerSave 2.0
- Qualcomm® Smart Transmit™ 2.0 technology
- Qualcomm® Wideband Envelope Tracking
- Qualcomm® AI-Enhanced Signal Boost
- Global 5G multi-SIM

Downlink: Up to 10 Gbps

Multimode support: 5G NR, LTE including CBRS, WCDMA, HSPA, TD-SCDMA, CDMA 1x, EV-DO, GSM/EDGE

Camera

Qualcomm Spectra™ Image Signal Processor
- Triple 18-bit ISPs
- Up to 3.2 Gigapixels per Second computer vision ISP (CV-ISP)
- Up to 36 MP triple camera @ 30 FPS with Zero Shutter Lag
- Up to 64+36 MP dual camera @ 30 FPS with Zero Shutter Lag
- Up to 108 MP single camera @ 30 FPS with Zero Shutter Lag
- Up to 200 Megapixel Photo Capture

Rec. 2020 color gamut photo and video capture

Up to 10-bit color depth photo and video capture

8K HDR Video Capture + 64 MP Photo Capture

10-bit HEIF: HEIC photo capture, HEVC video capture

Video Capture Formats: HDR10+, HDR10, HLG, Dolby Vision

8K HDR Video Capture @ 30 FPS

4K Video Capture @ 120 FPS

Slow-mo video capture at 720p @ 960 FPS

Bokeh Engine for Video Capture

Video super resolution

Multi-frame Noise Reduction (MFNR)

Locally Motion Compensated Temporal Filtering

Multi-Frame and triple exposure staggered/digital overlap HDR dual-sensor support

AI-based face detection, auto-focus, and auto-exposure

CPU

Kryo CPU
- Up to 3.0 GHz*, with Arm Cortex-X2 technology
- 64-bit Architecture

Visual Subsystem

Adreno GPU
- Vulkan® 1.1 API support
- HDR gaming (10-bit color depth, Rec. 2020 color gamut)
- Physically Based Rendering
- Volumetric Rendering
- Adreno Frame Motion Engine
- API Support: OpenGL® ES 3.2, OpenCL™ 2.0 FP, Vulkan 1.1
- Hardware-accelerated H.265 and VP9 decoder
- HDR Playback Codec support for HDR10+, HDR10, HLG and Dolby Vision

Security

Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU)

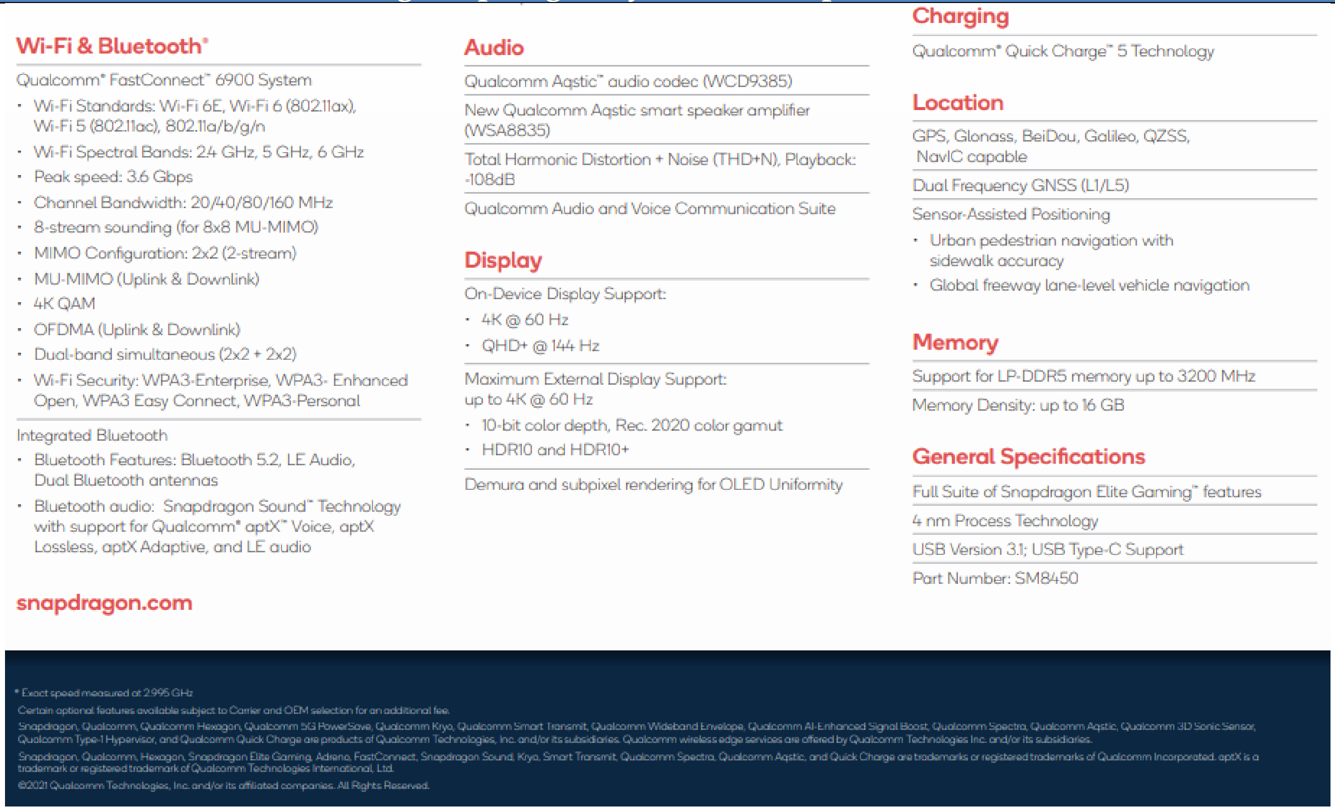
Trust Management Engine

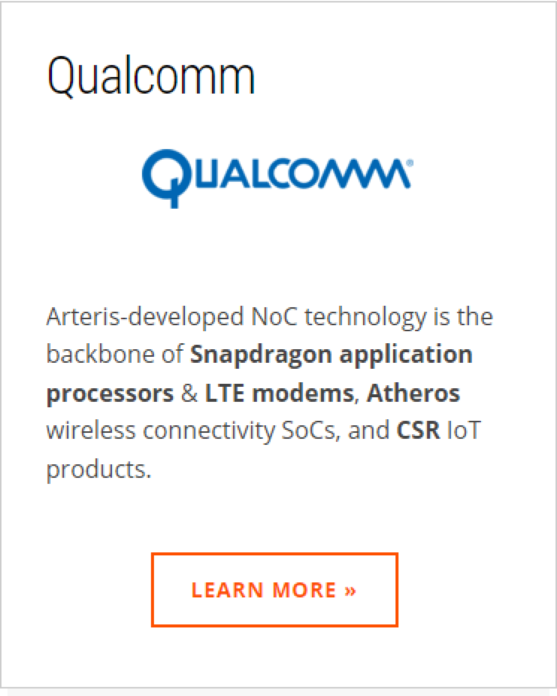
Qualcomm® wireless edge services (WES) and premium security features

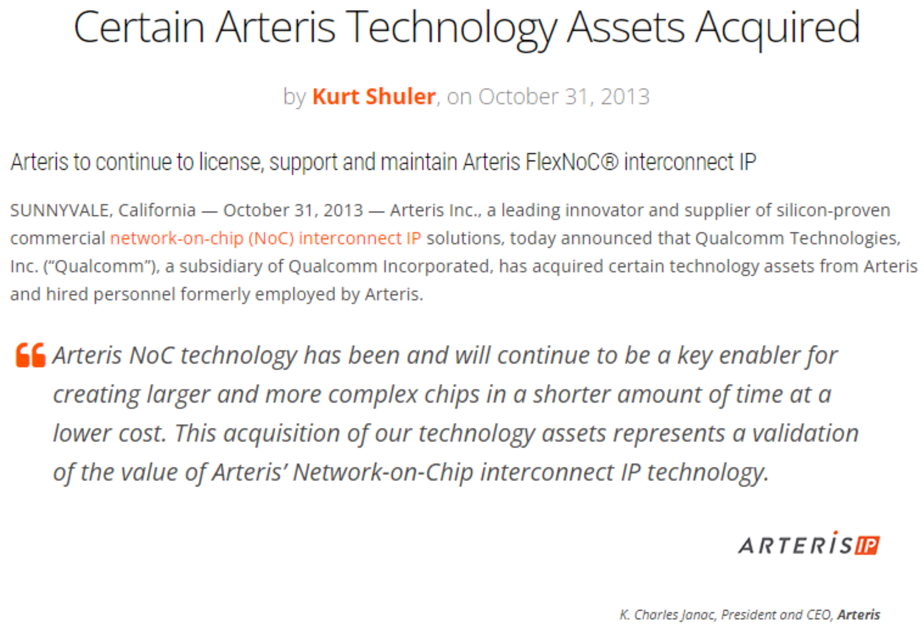
Qualcomm® 3D Sonic Sensor and Qualcomm 3D Sonic Max (fingerprint sensor)

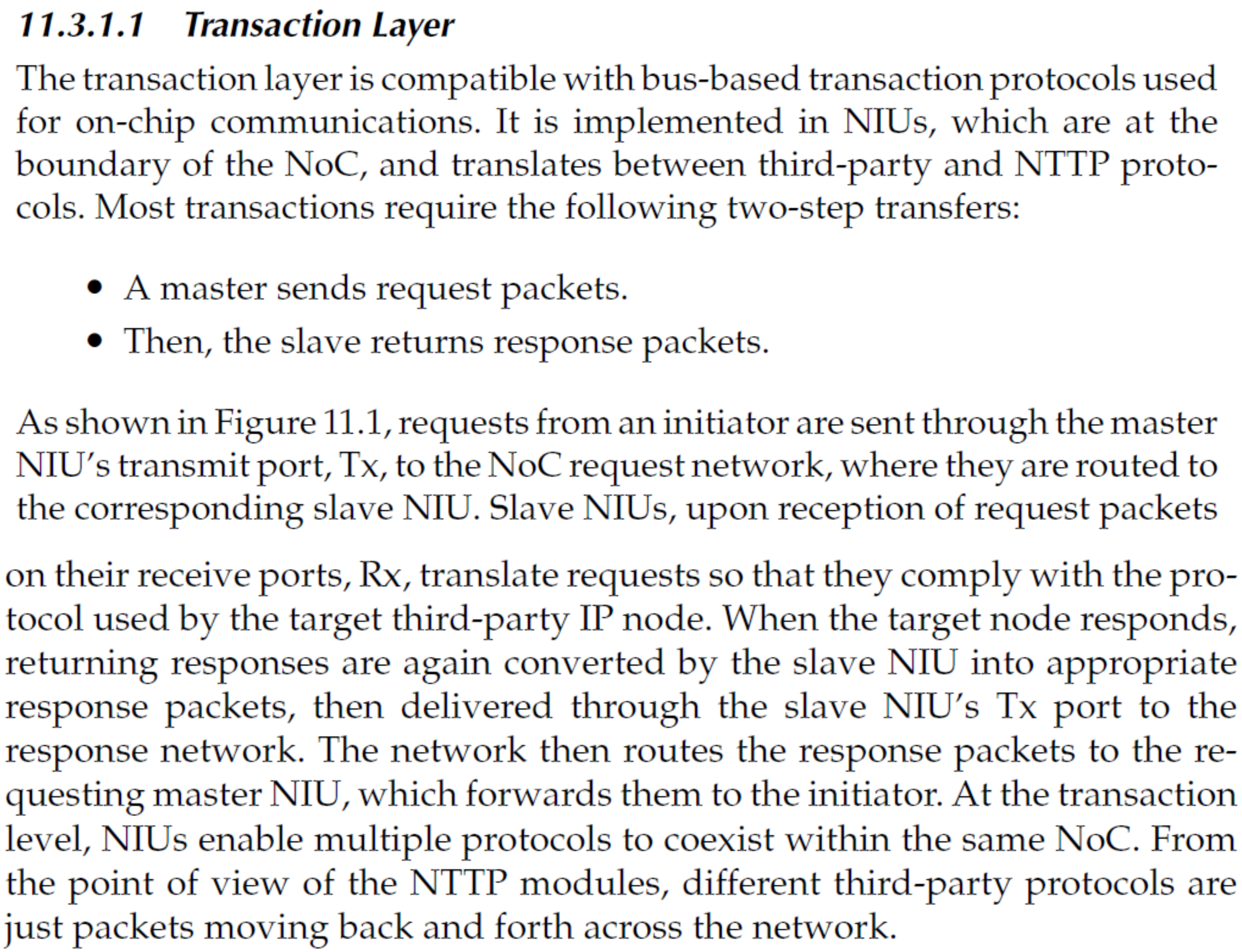
Qualcomm® Type-1 Hypervisor

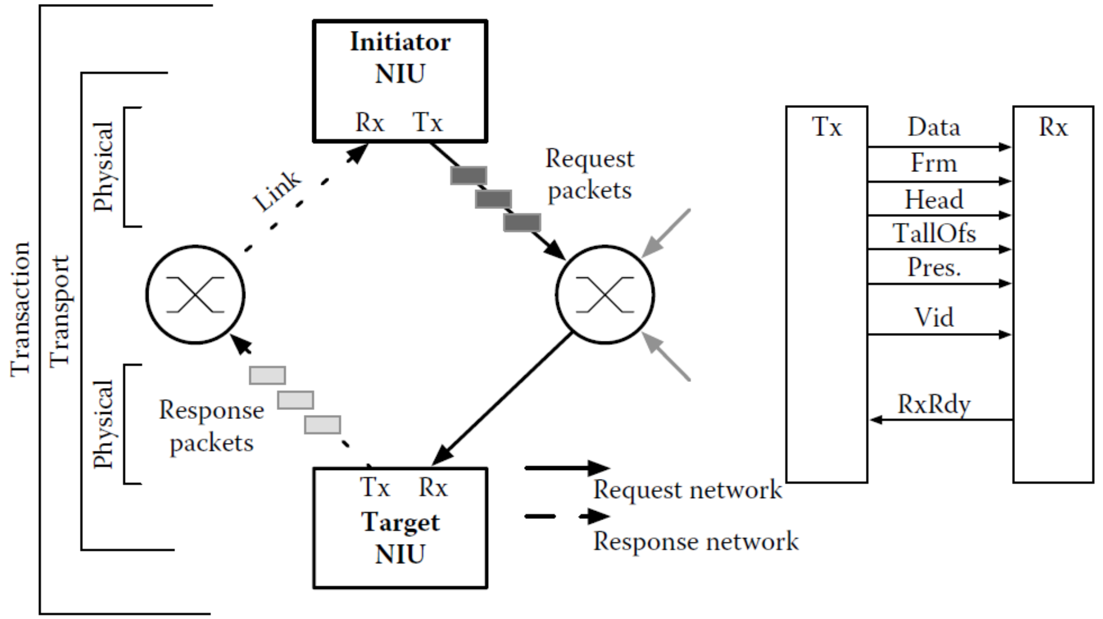
U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

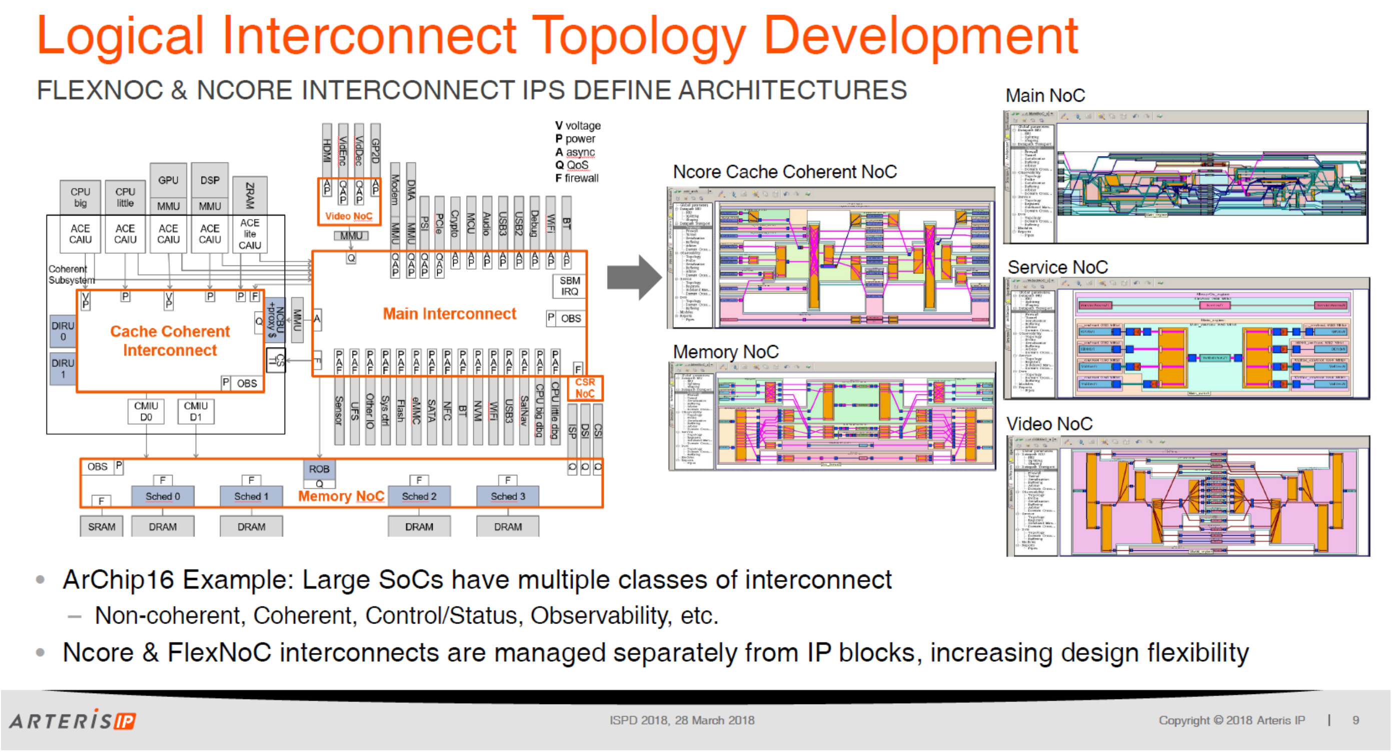
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **Wi-Fi & Bluetooth®** Qualcomm® FastConnect™ 6900 System • Wi-Fi Standards: Wi-Fi 6E, Wi-Fi 6 (802.11ax), Wi-Fi 5 (802.11ac), 802.11a/b/g/n • Wi-Fi Spectral Bands: 2.4 GHz, 5 GHz, 6 GHz • Peak speed: 3.6 Gbps • Channel Bandwidth: 20/40/80/160 MHz • 8-stream sounding (for 8x8 MU-MIMO) • MIMO Configuration: 2x2 (2-stream) • MU-MIMO (Uplink & Downlink) • 4K QAM • OFDMA (Uplink & Downlink) • Dual-band simultaneous (2x2 + 2x2) • Wi-Fi Security: WPA3-Enterprise, WPA3- Enhanced Open, WPA3 Easy Connect, WPA3-Personal Integrated Bluetooth • Bluetooth Features: Bluetooth 5.2, LE Audio, Dual Bluetooth antennas • Bluetooth audio: Snapdragon Sound™ Technology with support for Qualcomm® aptX™ Voice, aptX Lossless, aptX Adaptive, and LE audio snapdragon.com **Audio** Qualcomm Aqstic™ audio codec (WCD9385) New Qualcomm Aqstic smart speaker amplifier (WSA8835) Total Harmonic Distortion + Noise (THD+N), Playback: -108dB Qualcomm Audio and Voice Communication Suite **Display** On-Device Display Support: • 4K @ 60 Hz • QHD+ @ 144 Hz Maximum External Display Support: up to 4K @ 60 Hz • 10-bit color depth, Rec. 2020 color gamut • HDR10 and HDR10+ Demura and subpixel rendering for OLED Uniformity **Charging** Qualcomm® Quick Charge™ 5 Technology **Location** GPS, Glonass, BeiDou, Galileo, QZSS, NavIC capable Dual Frequency GNSS (L1/L5) Sensor-Assisted Positioning • Urban pedestrian navigation with sidewalk accuracy • Global freeway lane-level vehicle navigation **Memory** Support for LP-DDR5 memory up to 3200 MHz Memory Density: up to 16 GB **General Specifications** Full Suite of Snapdragon Elite Gaming™ features 4 nm Process Technology USB Version 3.1; USB Type-C Support Part Number: SM8450 * Exact speed measured at 2.995 GHz Certain optional features available subject to Carrier and OEM selection for an additional fee. Snapdragon, Qualcomm, Qualcomm Hexagon, Qualcomm 5G PowerSave, Qualcomm Kryo, Qualcomm Smart Transmit, Qualcomm Wideband Envelope, Qualcomm AI-Enhanced Signal Boost, Qualcomm Spectra, Qualcomm Aqstic, Qualcomm 3D Sonic Sensor, Qualcomm Type-1 Hypervisor, and Qualcomm Quick Charge are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm wireless edge services are offered by Qualcomm Technologies Inc. and/or its subsidiaries. Snapdragon, Qualcomm, Hexagon, Snapdragon Elite Gaming, Adreno, FastConnect, Snapdragon Sound, Kryo, Smart Transmit, Qualcomm Spectra, Qualcomm Aqstic, and Quick Charge are trademarks or registered trademarks of Qualcomm Incorporated. aptX is a trademark or registered trademark of Qualcomm Technologies International, Ltd. ©2021 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/snapdragon-8-gen-1-mobile-platform-product-brief.pdf The Snapdragon SoC included in the Motorola product utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the "Arteris NoC") for exchanging messages:

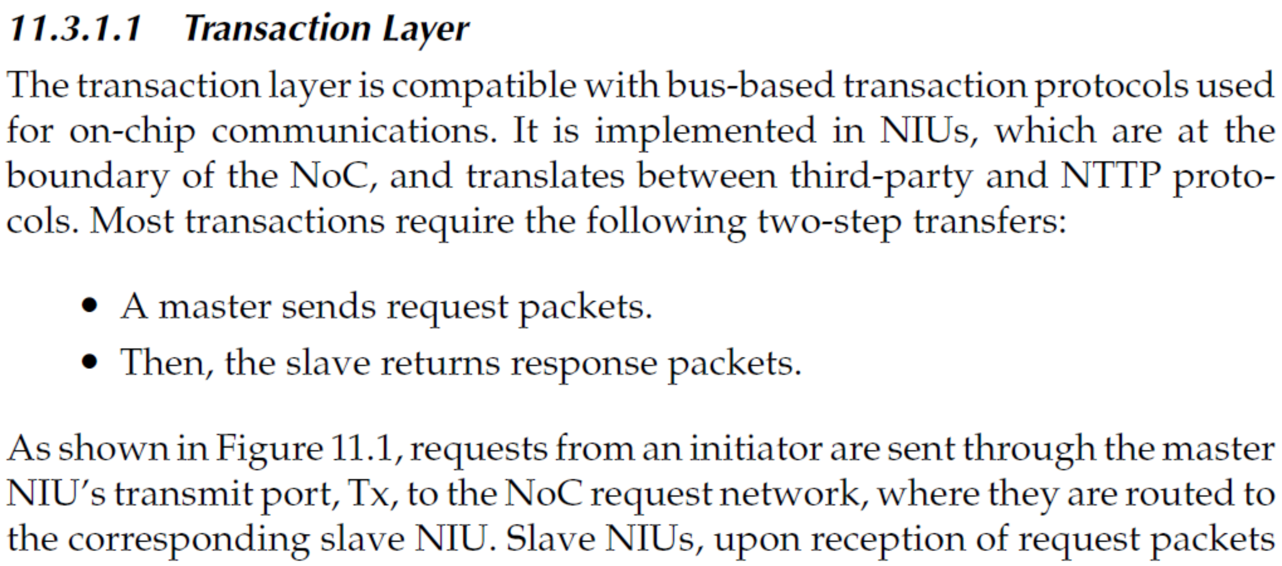
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Qualcomm Arteris-developed NoC technology is the backbone of **Snapdragon application processors** & **LTE modems**, **Atheros** wireless connectivity SoCs, and **CSR** IoT products. https://web.archive.org/web/20210514110614/https://www.arteris.com/customers

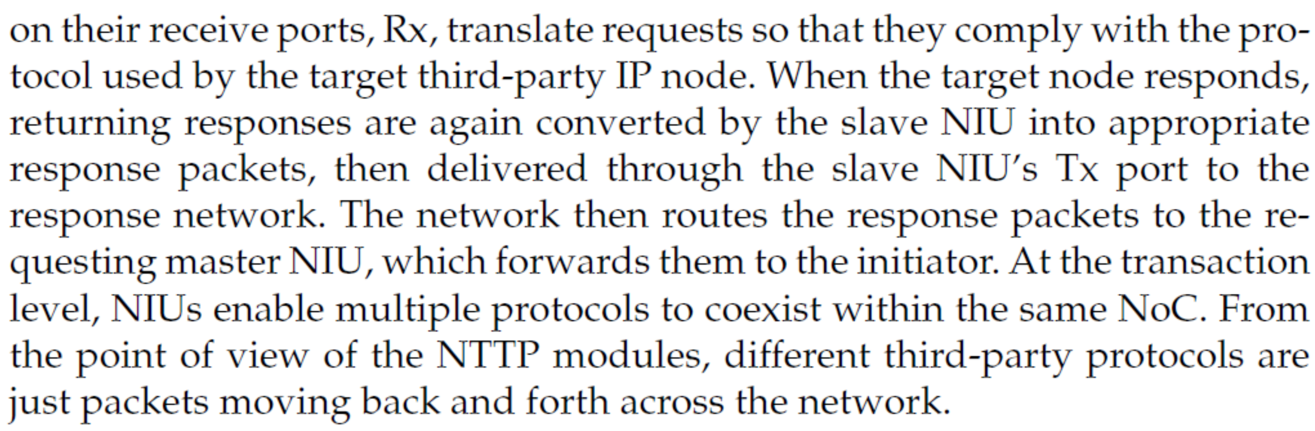
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Certain Arteris Technology Assets Acquired by **Kurt Shuler**, on October 31, 2013 Arteris to continue to license, support and maintain Arteris FlexNoC® interconnect IP SUNNYVALE, California — October 31, 2013 — Arteris Inc., a leading innovator and supplier of silicon-proven commercial network-on-chip (NoC) interconnect IP solutions, today announced that Qualcomm Technologies, Inc. ("Qualcomm"), a subsidiary of Qualcomm Incorporated, has acquired certain technology assets from Arteris and hired personnel formerly employed by Arteris. *Arteris NoC technology has been and will continue to be a key enabler for creating larger and more complex chips in a shorter amount of time at a lower cost. This acquisition of our technology assets represents a validation of the value of Arteris' Network-on-Chip interconnect IP technology.* *K. Charles Janac, President and CEO,* ***Arteris*** https://www.arteris.com/press-releases/Qualcomm-Arteris-asset-acquisition-2013_oct_31; https://www.fiercewireless.com/tech/qualcomm-acquires-arteris-noc-tech-assets-team The Arteris NoC exchanges messages between the plurality of modules via a network in the Snapdragon SoC included in the Motorola product. For example, in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":

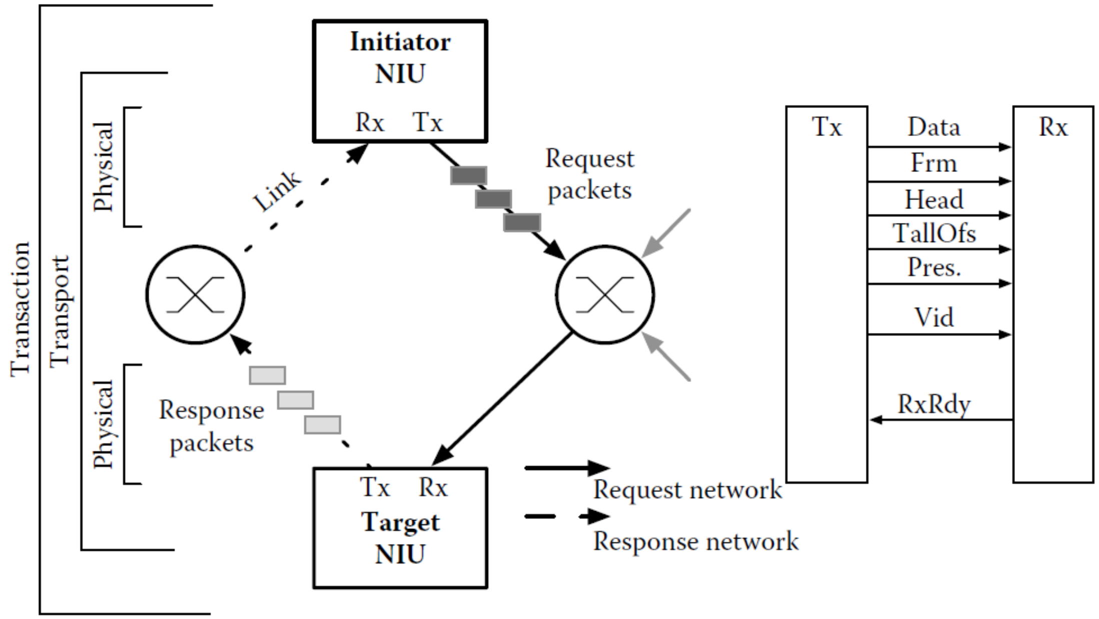
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: - A master sends request packets. - Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313; *see id* at 308 (explaining that Chapter 11 of this book describes the function of the Arteris NoC: "In this chapter we will present an MPSoC platform [...] using Arteris NoC as communication infrastructure."). As a further illustration, a large SoC, such as the Snapdragon SoC included in the Motorola product may include multiple classes of Arteris NoC network:

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 V voltage P power A async Q QoS F firewall CPU big CPU little GPU DSP MMU ZRAM ACE CAIU ACE lite CAIU Coherent Subsystem DIRU 0 DIRU 1 Cache Coherent Interconnect OBS CMIU D0 CMIU D1 NCBU +proxy $ HDMI VidEnc VidDec GP2D Video NoC Modem DMA PSI PCIe Crypto MCU Audio USB3 USB2 Debug WiFi BT SBM IRQ Main Interconnect CSR NoC Sensor UFS Other IO Sys ctrl Flash eMMC SATA NFC NVM SatNav CPU big dbg CPU little dbg ISP DSI CSI ROB Memory NoC Sched 0 Sched 1 Sched 2 Sched 3 SRAM DRAM Ncore Cache Coherent NoC Main NoC Service NoC Video NoC **Logical Interconnect Topology Development** FLEXNOC & NCORE INTERCONNECT IPS DEFINE ARCHITECTURES - ArChip16 Example: Large SoCs have multiple classes of interconnect – Non-coherent, Coherent, Control/Status, Observability, etc. - Ncore & FlexNoC interconnects are managed separately from IP blocks, increasing design flexibility ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 9 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9.
wherein a message issued by an addressing module M comprises:	Without conceding that the preamble of claim 4 of the '9893 Patent is limiting, a message issued by an addressing module M in the Snapdragon SoC included in the Motorola product via the Arteris NoC comprises first information indicative of a location of an addressed message receiving module S within the network and is comprised of (1) a connection identifier identifying two or more message receiving modules S and (2) an identifier of a passive network interface means associated with the addressed message receiving module S, and second information

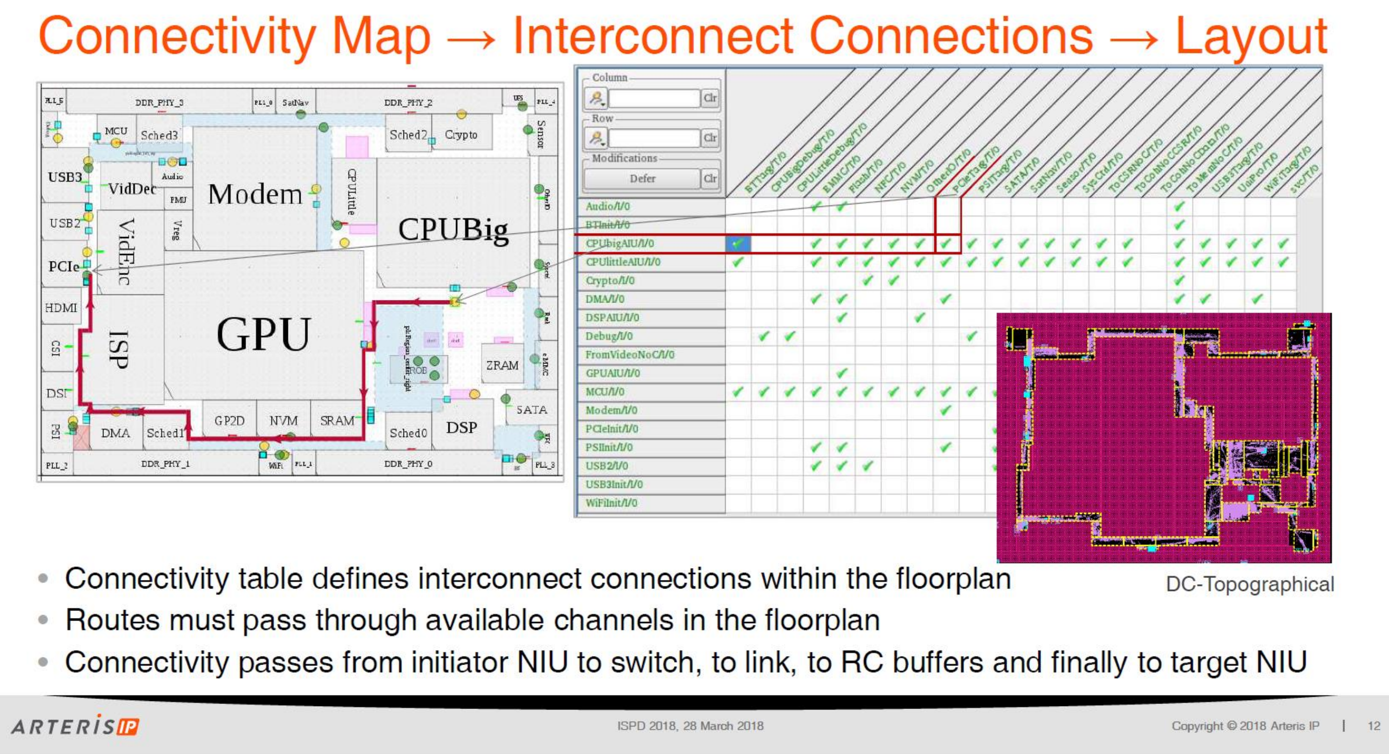
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
first information indicative of a location of an addressed message receiving module S within the network and is comprised of (1) a connection identifier identifying two or more message receiving modules S and (2) an identifier of a passive network interface means associated with the addressed message receiving module S, and second information indicative of a particular location within the addressed message receiving module S, such as	indicative of a particular location within the addressed message receiving module S, such as a memory, or a register address, either literally or under the doctrine of equivalents. For example, the Arteris NoC used in the Snapdragon SoC included in the Motorola product uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  *11.3.1.1 Transaction Layer* The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
a memory, or a register address,	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. As a further illustration, connections between initiator module NIUs (e.g., "CPUbigAIU/1/0") and two or more target module NIUs (e.g., "ETTarg/T/0," "EMMC/T/0," "Flash/T/0," "NFC/T0," "PCIeTarg/T/0," etc.) within the Arteris NoC may be defined by a connectivity table:

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]



Connectivity Map → Interconnect Connections → Layout

	BTTarg/T/0	CPUBigDebug/T/0	CPULittleDebug/T/0	EMMC/T/0	Flash/T/0	NFC/T/0	NVM/T/0	OtherIO/T/0	PCIeTarg/T/0	PSITarg/T/0	SATA/T/0	SatNav/T/0	Sensor/T/0	SysCtrl/T/0	To CSRNoC/T/0	To CohNoCCSR/T/0	To CohNoCData/T/0	To MemNoC/T/0	USB3Targ/T/0	UniPro/T/0	WiFiTarg/T/0	svc/T/0
Audio/I/0				✓	✓													✓				
BTInit/I/0																		✓				
CPUbigAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CPUlittleAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Crypto/I/0						✓	✓											✓				
DMA/I/0				✓	✓				✓									✓	✓		✓	
DSPAIU/I/0					✓			✓														
Debug/I/0		✓	✓							✓												
FromVideoNoC/I/0																						
GPUAIU/I/0					✓																	
MCU/I/0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓											
Modem/I/0									✓													
PCIeInit/I/0											✓											
PSIInit/I/0				✓	✓				✓		✓											
USB2/I/0				✓	✓	✓					✓											
USB3Init/I/0																						
WiFiInit/I/0																						

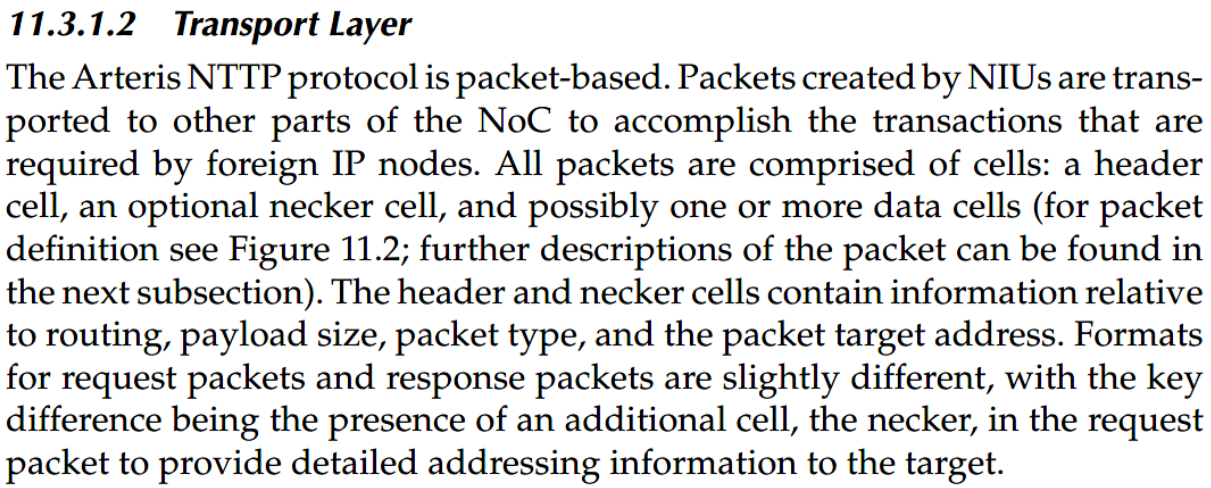
DC-Topographical

- Connectivity table defines interconnect connections within the floorplan
- Routes must pass through available channels in the floorplan
- Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU

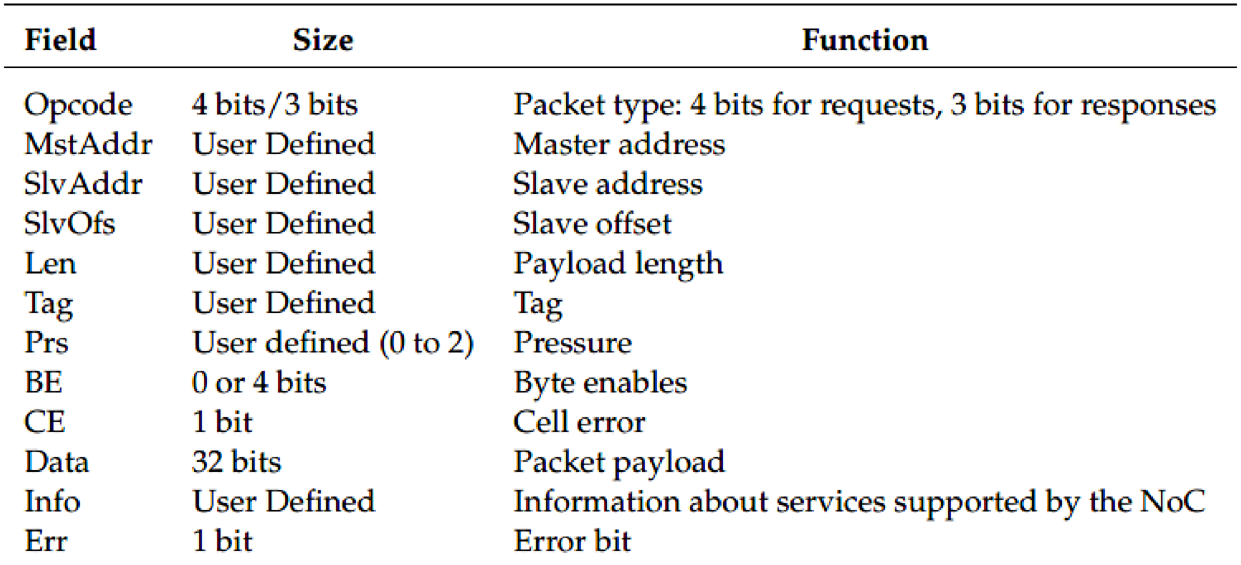
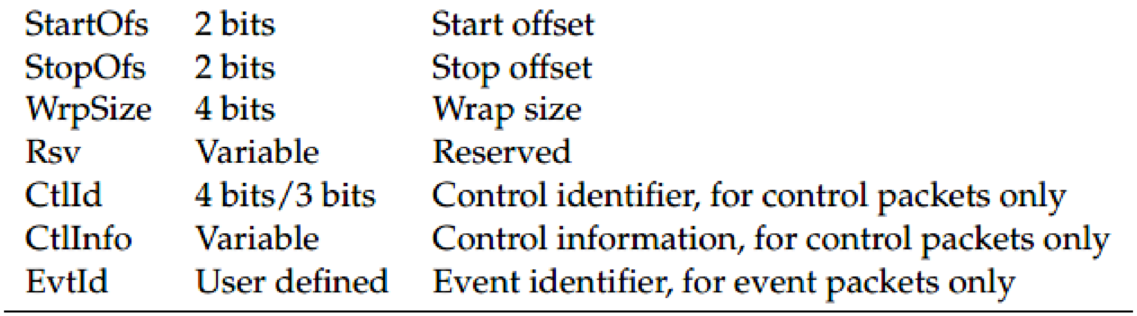
ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 12

See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12.

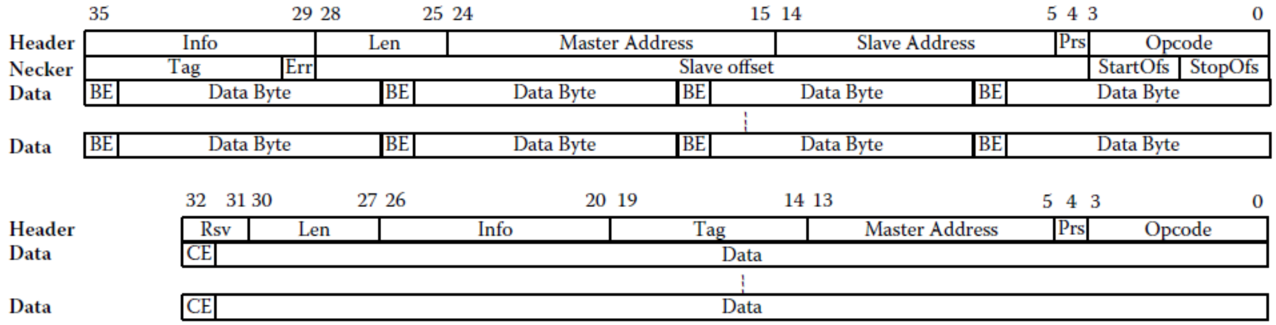
As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":

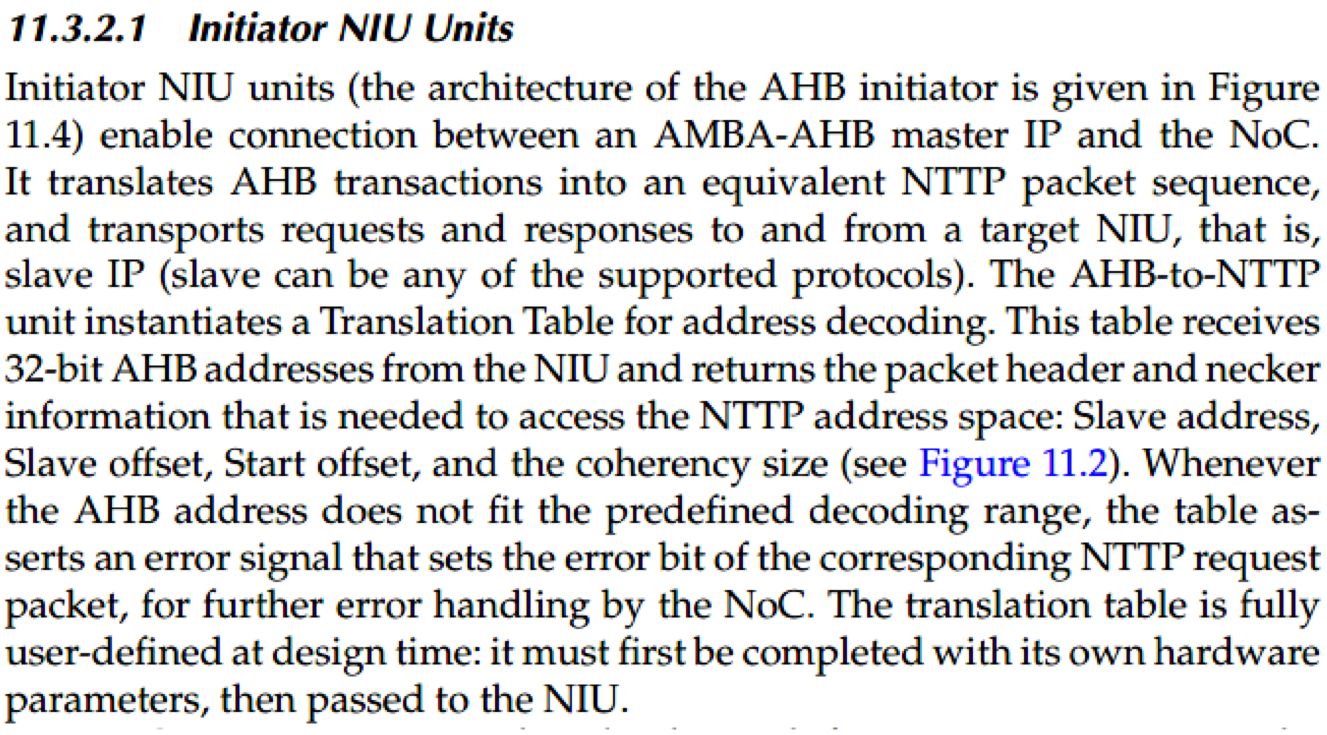
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313. As a further example, the packets sent in the Arteris NoC are "composed of cells that are organized into fields, with each field carrying specific information," including "Slave address" and "Slave offset":

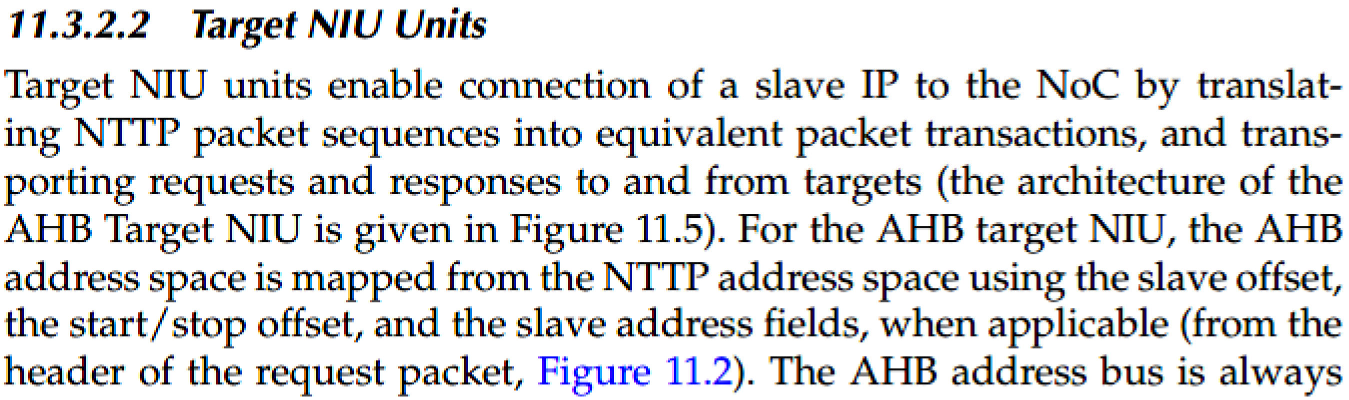
U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

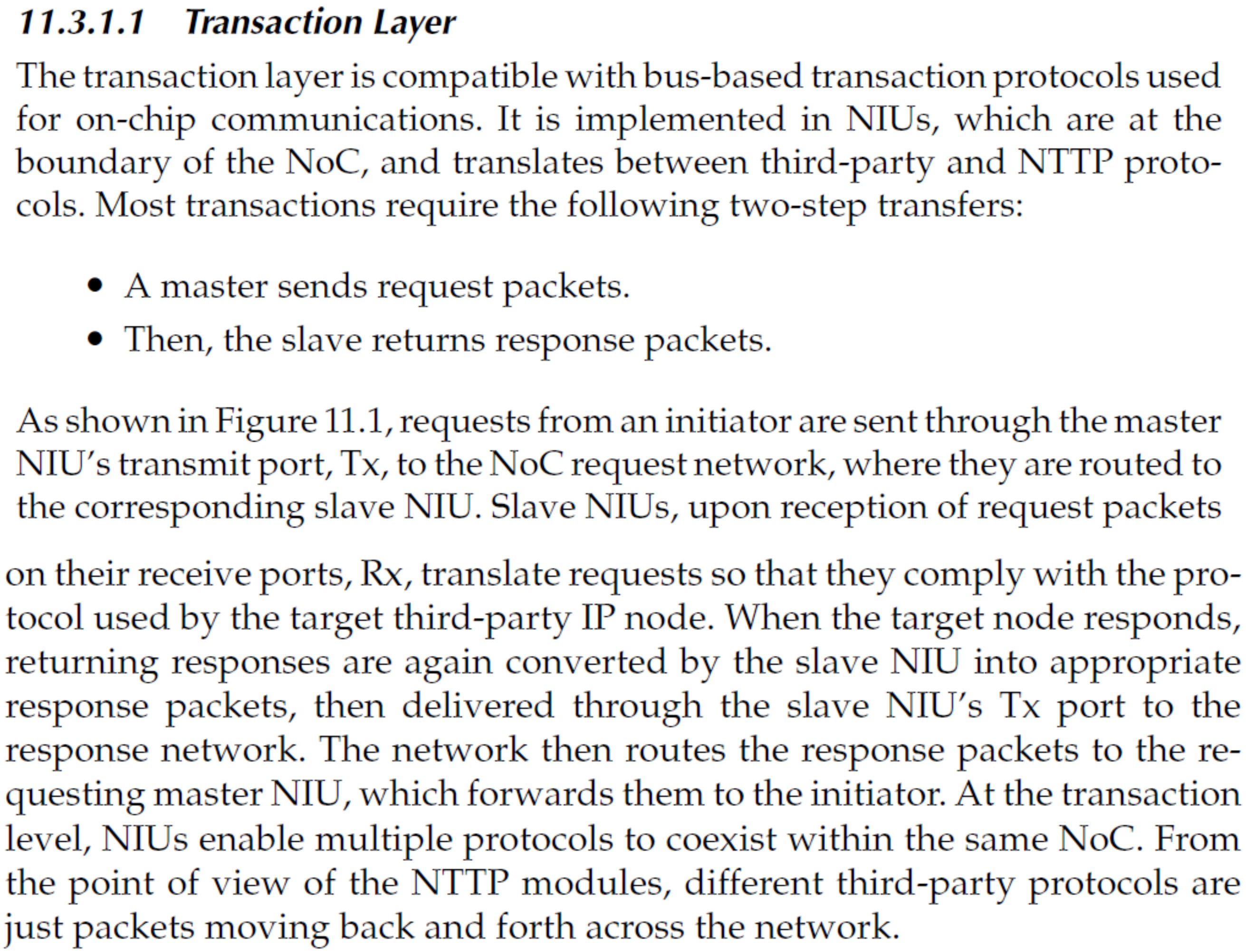
<table>
<tr><th>'9893 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip[1]</th></tr>
<tr><td></td><td>
<table>
<tr><th>Field</th><th>Size</th><th>Function</th></tr>
<tr><td>Opcode</td><td>4 bits/3 bits</td><td>Packet type: 4 bits for requests, 3 bits for responses</td></tr>
<tr><td>MstAddr</td><td>User Defined</td><td>Master address</td></tr>
<tr><td>SlvAddr</td><td>User Defined</td><td>Slave address</td></tr>
<tr><td>SlvOfs</td><td>User Defined</td><td>Slave offset</td></tr>
<tr><td>Len</td><td>User Defined</td><td>Payload length</td></tr>
<tr><td>Tag</td><td>User Defined</td><td>Tag</td></tr>
<tr><td>Prs</td><td>User defined (0 to 2)</td><td>Pressure</td></tr>
<tr><td>BE</td><td>0 or 4 bits</td><td>Byte enables</td></tr>
<tr><td>CE</td><td>1 bit</td><td>Cell error</td></tr>
<tr><td>Data</td><td>32 bits</td><td>Packet payload</td></tr>
<tr><td>Info</td><td>User Defined</td><td>Information about services supported by the NoC</td></tr>
<tr><td>Err</td><td>1 bit</td><td>Error bit</td></tr>
</table>

<table>
<tr><th></th><th></th><th></th></tr>
<tr><td>StartOfs</td><td>2 bits</td><td>Start offset</td></tr>
<tr><td>StopOfs</td><td>2 bits</td><td>Stop offset</td></tr>
<tr><td>WrpSize</td><td>4 bits</td><td>Wrap size</td></tr>
<tr><td>Rsv</td><td>Variable</td><td>Reserved</td></tr>
<tr><td>CtlId</td><td>4 bits/3 bits</td><td>Control identifier, for control packets only</td></tr>
<tr><td>CtlInfo</td><td>Variable</td><td>Control information, for control packets only</td></tr>
<tr><td>EvtId</td><td>User defined</td><td>Event identifier, for event packets only</td></tr>
</table></td></tr>
</table>

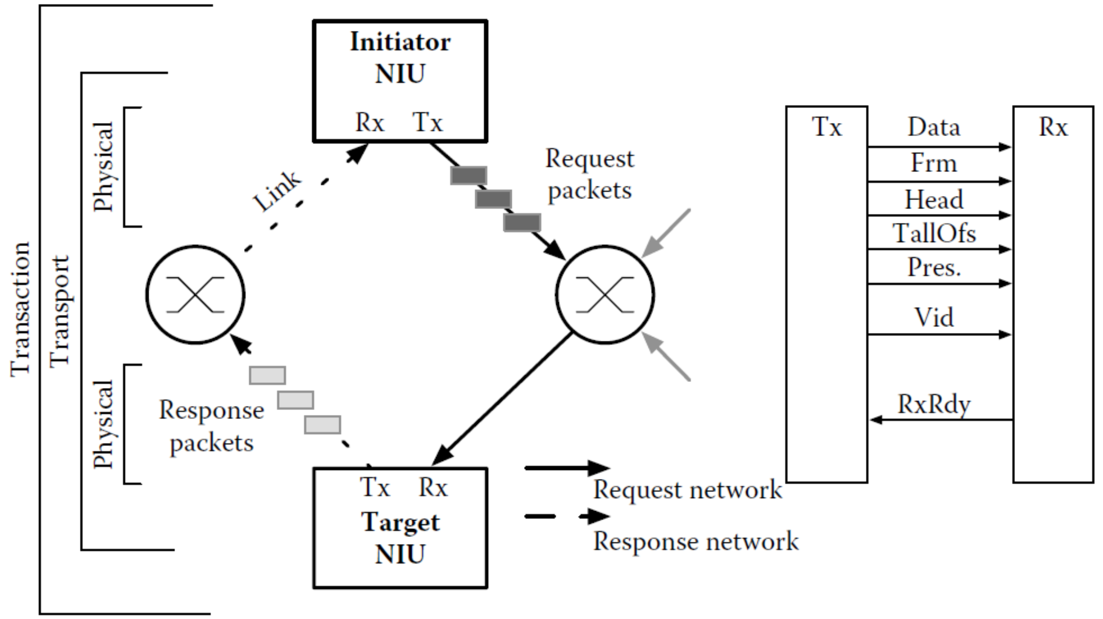
U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 35 29 28 25 24 15 14 5 4 3 0 Header Info Len Master Address Slave Address Prs Opcode Necker Tag Err Slave offset StartOfs StopOfs Data BE Data Byte BE Data Byte BE Data Byte BE Data Byte Data BE Data Byte BE Data Byte BE Data Byte BE Data Byte 32 31 30 27 26 20 19 14 13 5 4 3 0 Header Rsv Len Info Tag Master Address Prs Opcode Data CE Data Data CE Data **FIGURE 11.2** NTTP packet structure. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 314-315. As a further example, "[i]nitiator NIU units…translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and the "AHB-to-NTTP unit instantiates a Translation Table for address decoding" with the table "receiv[ing] 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size":

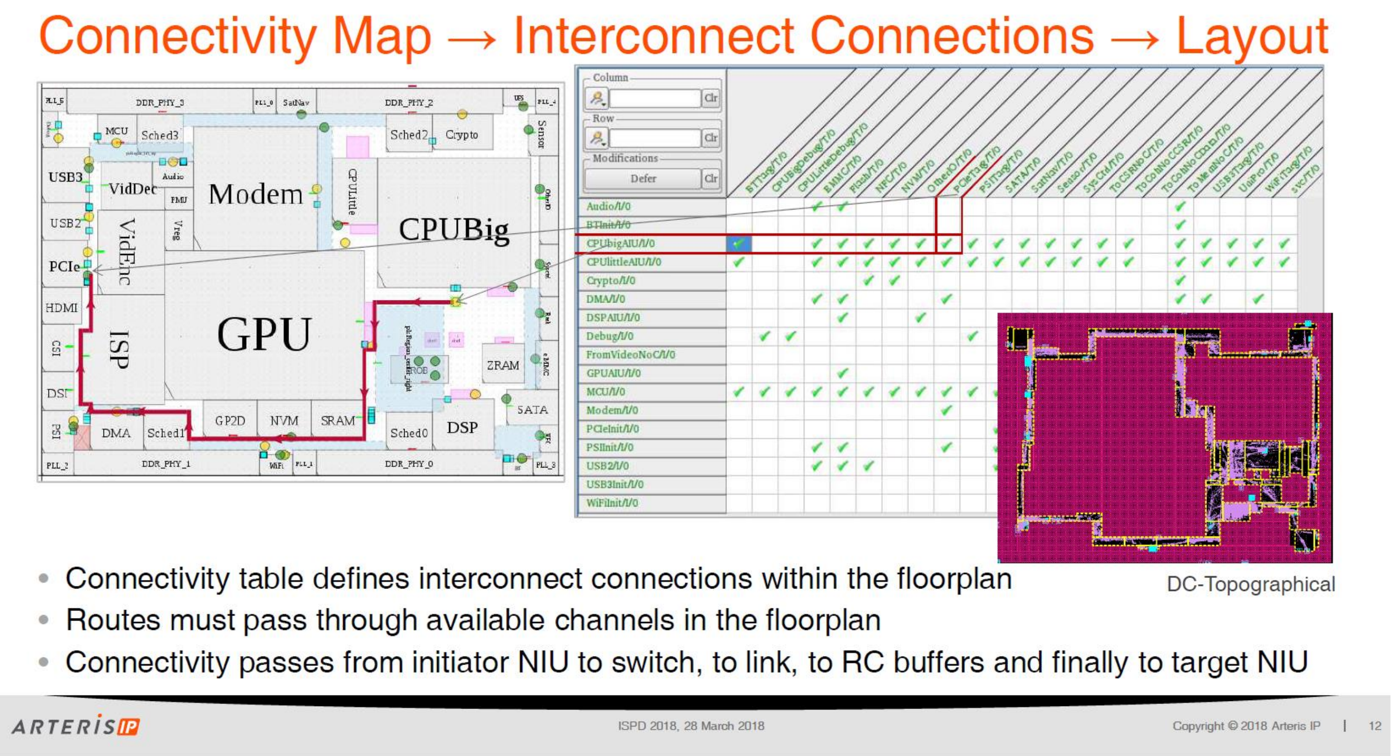
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **11.3.2.1 Initiator NIU Units** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. As further example, "[f]or the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2)":

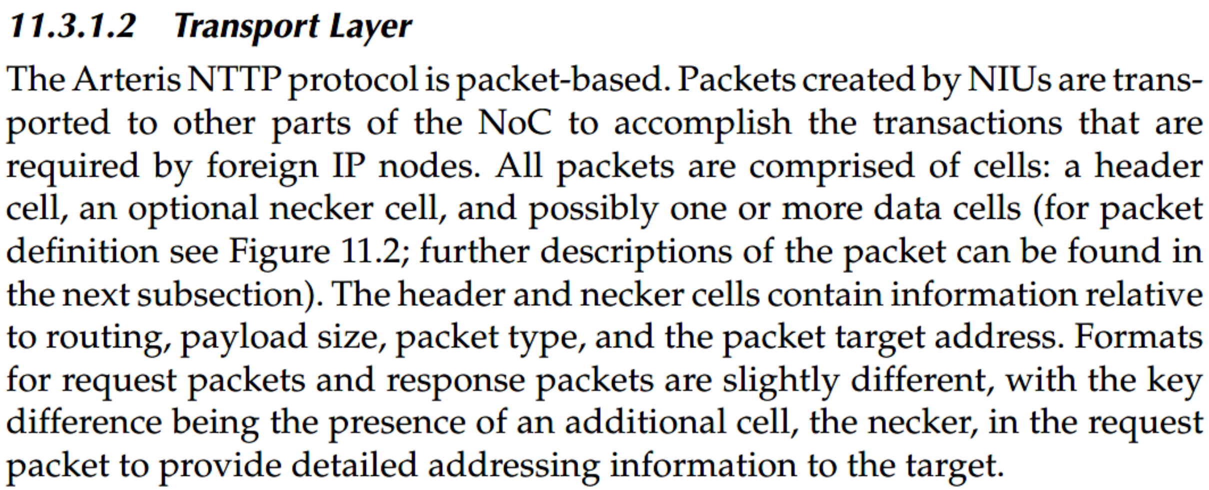
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **11.3.2.2 Target NIU Units** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always *Id.* at 318.
the method including the steps of: (a) issuing from said addressing module M a message request including said first information, said second information, and data and/or connection properties to an address	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product issues from said addressing module M a message request including said first information, said second information, and data and/or connection properties to an address translation unit included as part of an active network interface module associated with said addressing module M, either literally or under the doctrine of equivalents. For example, the Arteris NoC used in the Snapdragon SoC included in the Motorola product uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
translation unit included as part of an active network interface module associated with said addressing module M,	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. As a further illustration, connections between initiator module NIUs (e.g., "CPUbigAIU/1/0") and two or more target module NIUs (e.g., "ETTarg/T/0," "EMMC/T/0," "Flash/T/0," "NFC/T0," "PCIeTarg/T/0," etc.) within the Arteris NoC may be defined by a connectivity table:

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Connectivity Map → Interconnect Connections → Layout - Connectivity table defines interconnect connections within the floorplan - Routes must pass through available channels in the floorplan - Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU DC-Topographical ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 12 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12. As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":

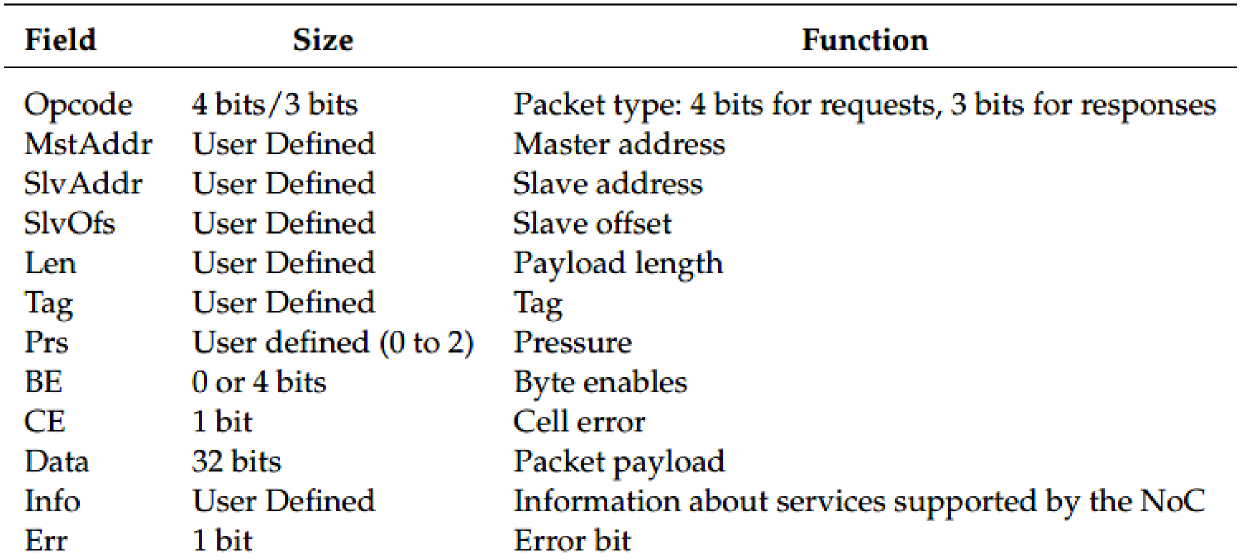
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As yet a further illustration, packets in the Arteris NoC are "delivered as words that are sent along links and "[o]ne link (represented in Figure 11.1) defines the following signals," which include "the current priority of the packet used to define preferred traffic class (or Quality of Service)" and "[f]low control":

maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in Figure 11.1) defines the following signals:

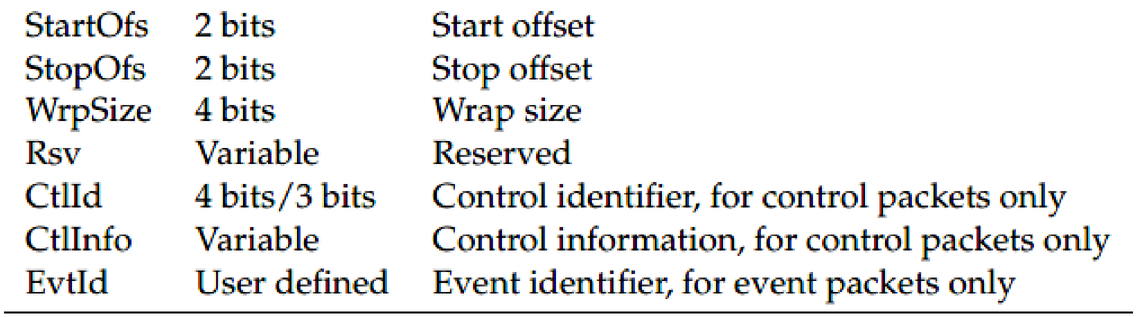
- **Data**—Data word of the width specified at design-time.
- **Frm**—When asserted high, indicates that a packet is being transmitted.
- **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only.
- **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only.
- **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2).
- **Vld**—Data valid: when asserted high, indicates that a word is being transmitted.
- **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy.

This signal set, which constitutes the Media Independent NoC Interface (MINI), is the foundation for NTTP communications.

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

<table>
<tr><th>'9893 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip[1]</th></tr>
<tr><td></td><td>
<p><i>Id.</i> at 313-314.</p>
<p>As a further example, the packets sent in the Arteris NoC are "composed of cells that are organized into fields, with each field carrying specific information," including "Pres," "Slave address" and "Slave offset":</p>

<table>
<tr><th>Field</th><th>Size</th><th>Function</th></tr>
<tr><td>Opcode</td><td>4 bits/3 bits</td><td>Packet type: 4 bits for requests, 3 bits for responses</td></tr>
<tr><td>MstAddr</td><td>User Defined</td><td>Master address</td></tr>
<tr><td>SlvAddr</td><td>User Defined</td><td>Slave address</td></tr>
<tr><td>SlvOfs</td><td>User Defined</td><td>Slave offset</td></tr>
<tr><td>Len</td><td>User Defined</td><td>Payload length</td></tr>
<tr><td>Tag</td><td>User Defined</td><td>Tag</td></tr>
<tr><td>Prs</td><td>User defined (0 to 2)</td><td>Pressure</td></tr>
<tr><td>BE</td><td>0 or 4 bits</td><td>Byte enables</td></tr>
<tr><td>CE</td><td>1 bit</td><td>Cell error</td></tr>
<tr><td>Data</td><td>32 bits</td><td>Packet payload</td></tr>
<tr><td>Info</td><td>User Defined</td><td>Information about services supported by the NoC</td></tr>
<tr><td>Err</td><td>1 bit</td><td>Error bit</td></tr>
</table>
</td></tr>
</table>

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]



StartOfs	2 bits	Start offset
StopOfs	2 bits	Stop offset
WrpSize	4 bits	Wrap size
Rsv	Variable	Reserved
CtlId	4 bits/3 bits	Control identifier, for control packets only
CtlInfo	Variable	Control information, for control packets only
EvtId	User defined	Event identifier, for event packets only

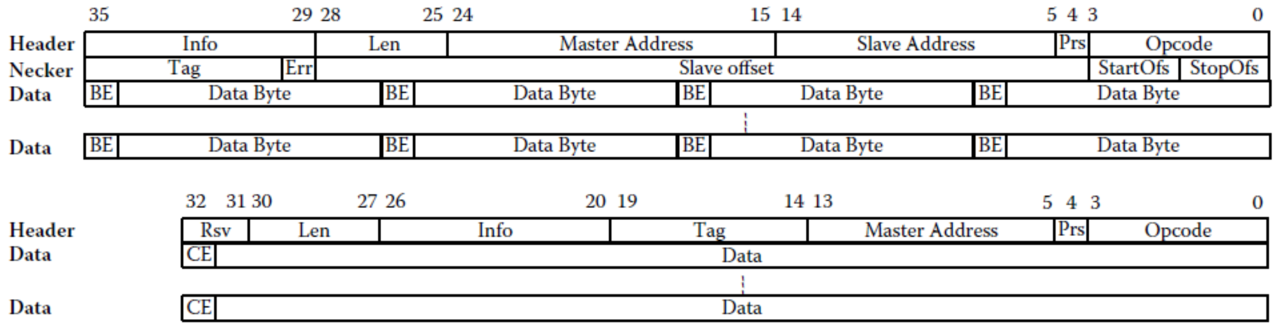
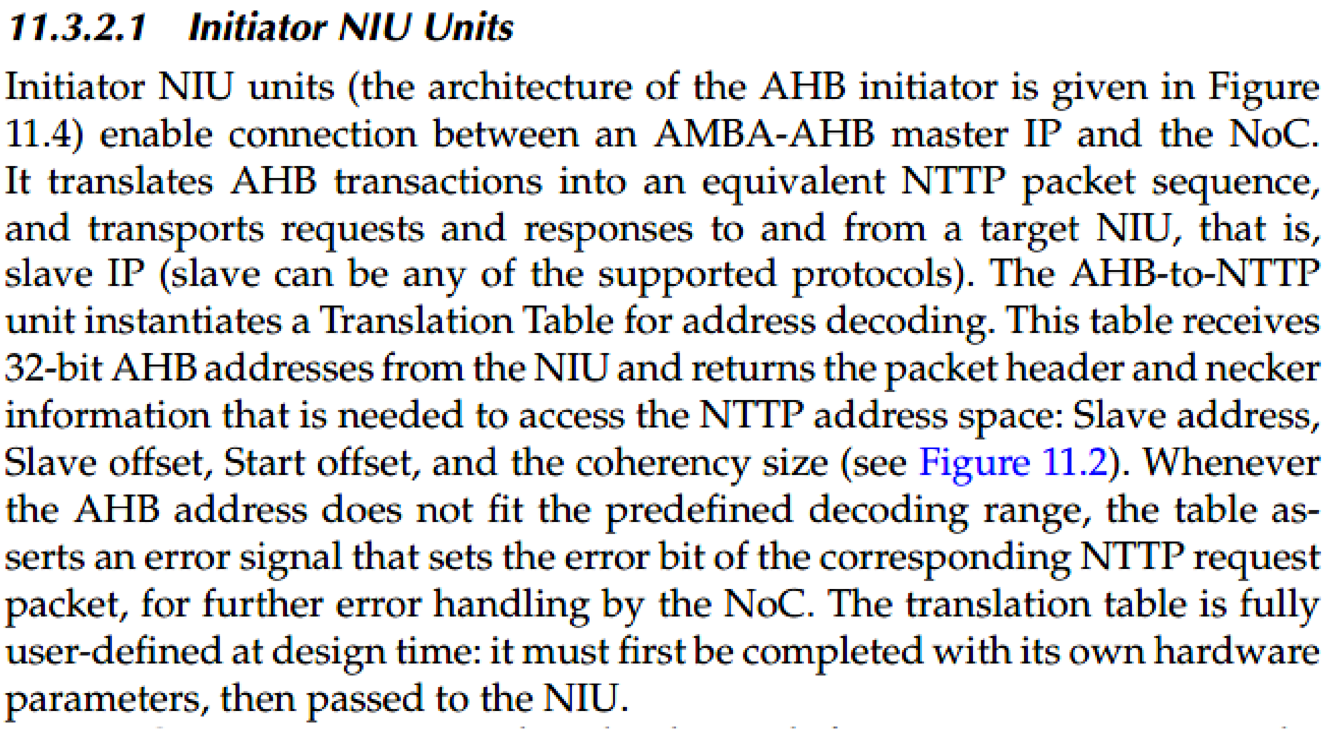
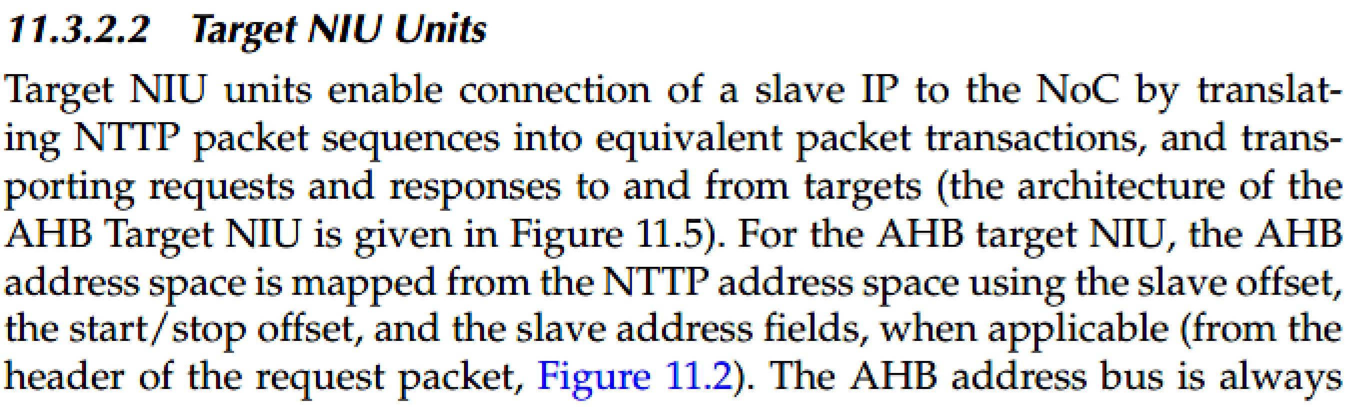


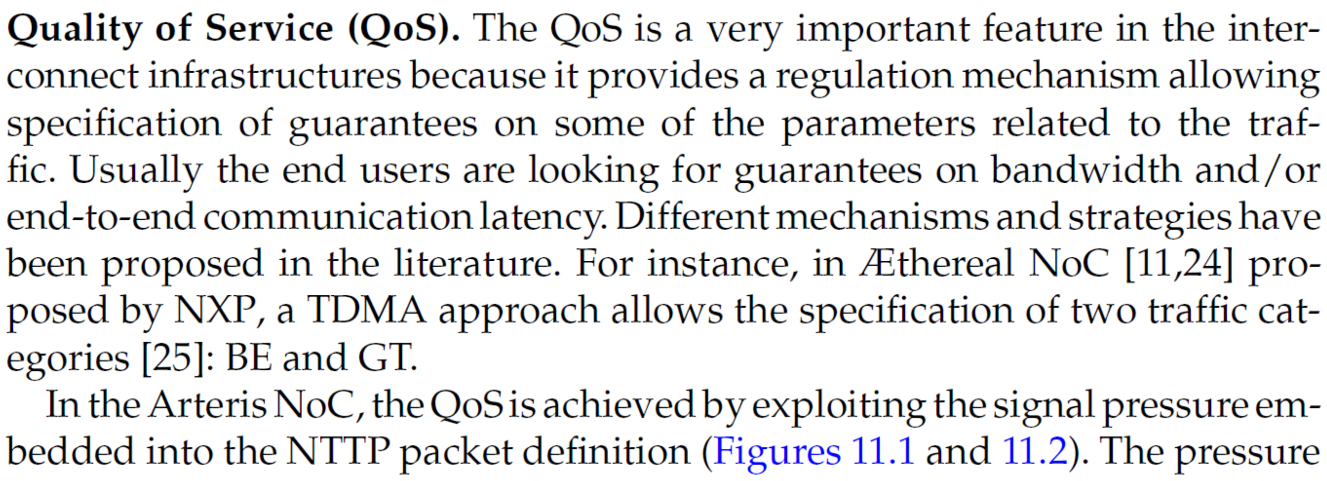
FIGURE 11.2
NTTP packet structure.

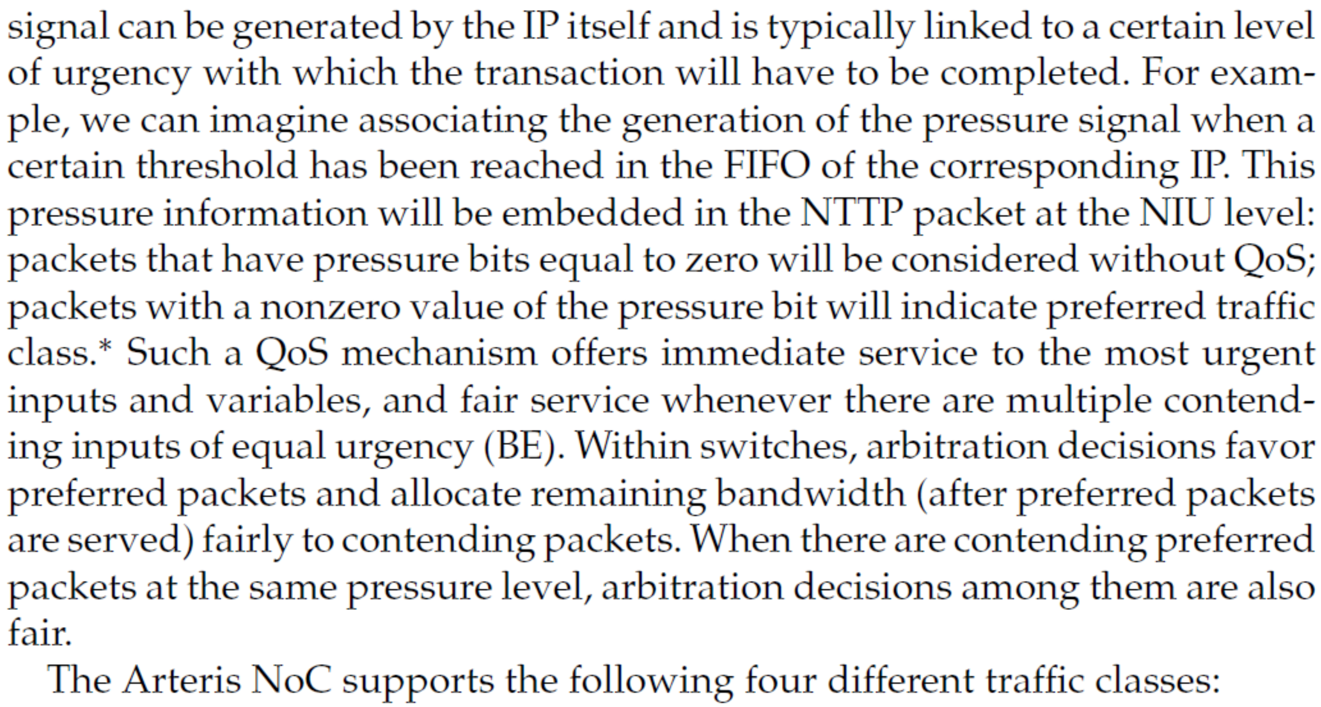
Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 314-315.

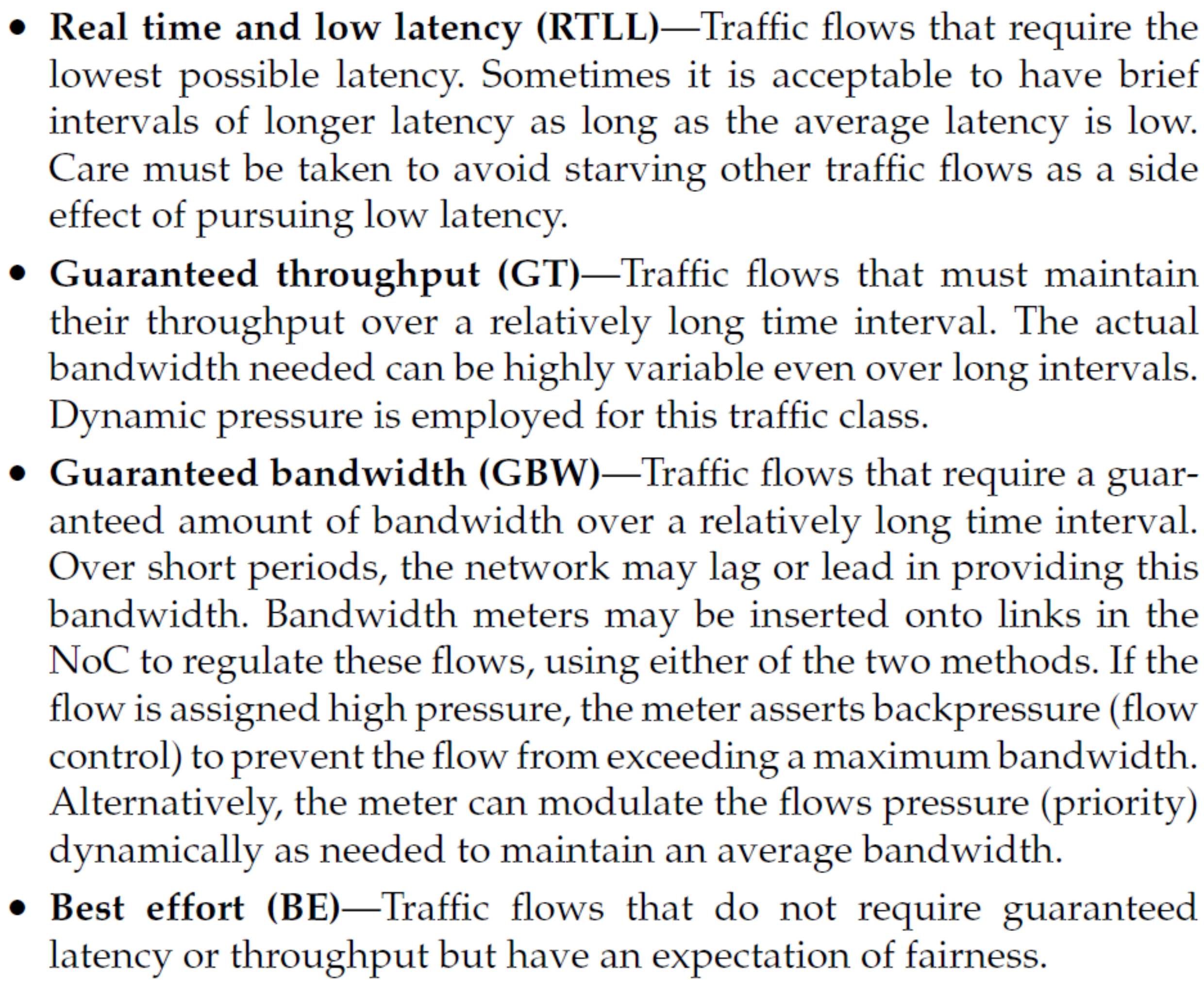
As a further example, "[i]nitiator NIU units…translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target

’9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	NIU, that is, slave IP” and the “AHB-to-NTTP unit instantiates a Translation Table for address decoding” with the table “receiv[ing] 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size”:  ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	As further example, "[f]or the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2)":  ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always *Id.* at 318. As a further illustration, the Arteris NoC implements Quality of Service (QoS) to "provide[] a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic"; "QoS, which includes guarantees of throughput and/or latency, is achieved by exploiting the signal pressure embedded into the NTTP packet definition" where the "pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed"; and the "pressure information will be embedded in the NTTP packet at the NIU level":

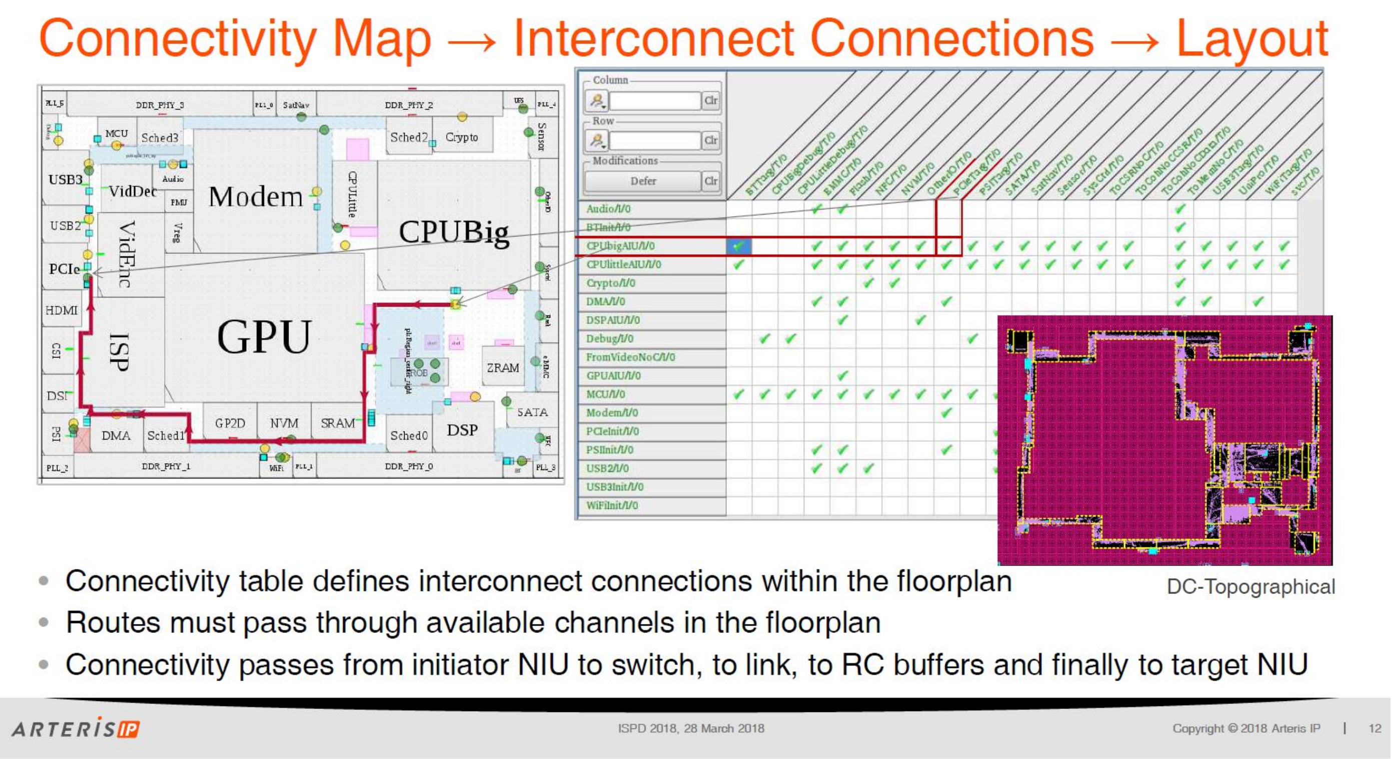
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 - **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. - **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. - **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. - **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316. Connections within the Arteris NoC may be defined by a connectivity table:



Connectivity Map → Interconnect Connections → Layout

	BTTarg/T/0	CPUBigDebug/T/0	CPULittleDebug/T/0	EMMC/T/0	Flash/T/0	NFC/T/0	NVM/T/0	OtherIO/T/0	PCIeTarg/T/0	PSITarg/T/0	SATA/T/0	SatNav/T/0	Sensor/T/0	SysCrd/T/0	ToCSRNoC/T/0	ToCohNoCCSR/T/0	ToCohNoCData/T/0	ToMemNoC/T/0	USB3Targ/T/0	UniPro/T/0	WiFiTarg/T/0	svc/T/0
Audio/I/0				✓	✓													✓				
BTInit/I/0																		✓				
CPUbigAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CPUlittleAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Crypto/I/0						✓	✓											✓				
DMA/I/0				✓	✓				✓									✓	✓		✓	
DSPAIU/I/0					✓			✓														
Debug/I/0		✓	✓							✓												
FromVideoNoC/I/0																						
GPUAIU/I/0					✓																	
MCU/I/0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓											
Modem/I/0									✓													
PCIeInit/I/0											✓											
PSIInit/I/0				✓	✓				✓		✓											
USB2/I/0				✓	✓	✓					✓											
USB3Init/I/0																						
WiFiInit/I/0																						

- Connectivity table defines interconnect connections within the floorplan
- Routes must pass through available channels in the floorplan
- Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU

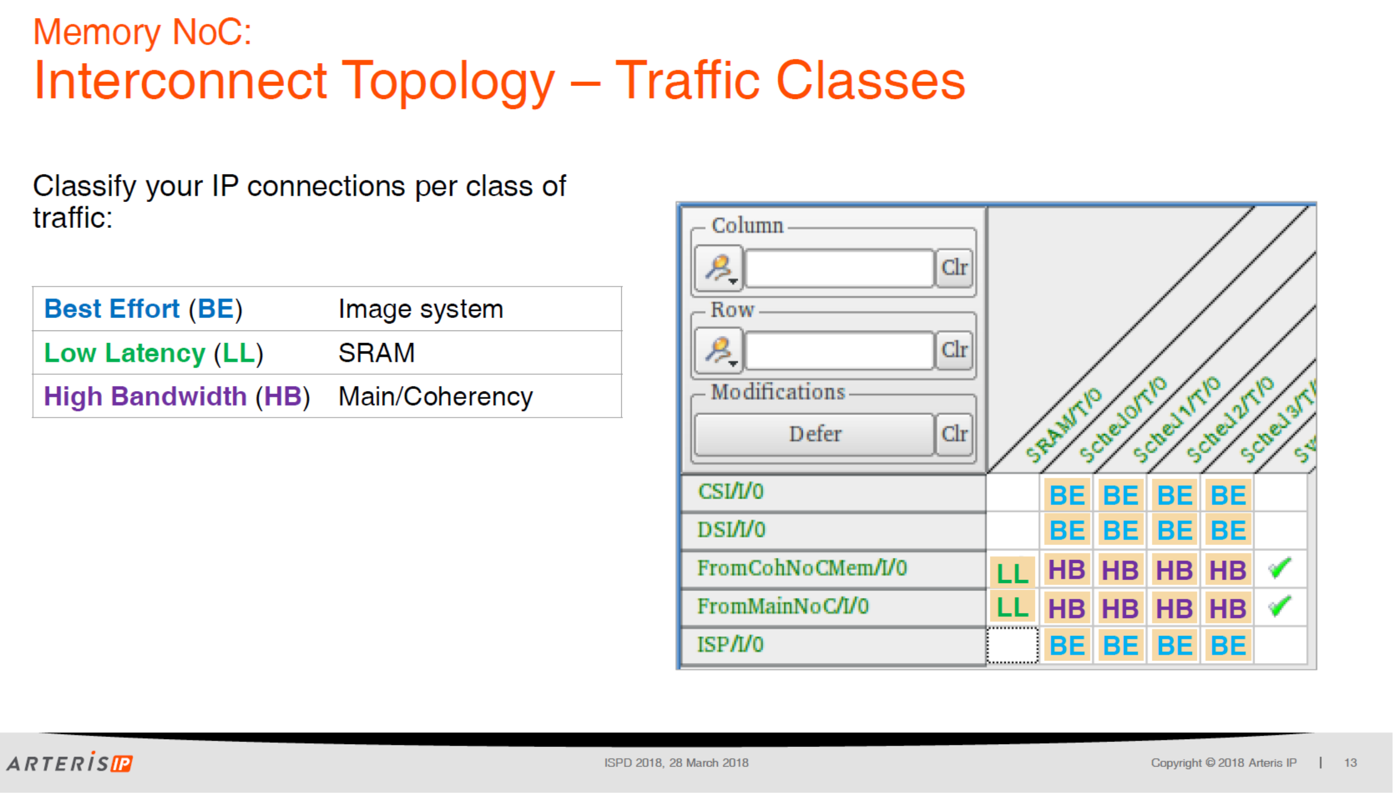
DC-Topographical

ARTERIS IP

ISPD 2018, 28 March 2018

Copyright © 2018 Arteris IP | 12

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12. As a further illustration, connections within the Arteris NoC may be classified by traffic class and traffic classes, including related to latency, may be mapped onto the Arteris interconnect topology:



Memory NoC:
Interconnect Topology – Traffic Classes

Classify your IP connections per class of traffic:

Best Effort (BE)	Image system
Low Latency (LL)	SRAM
High Bandwidth (HB)	Main/Coherency

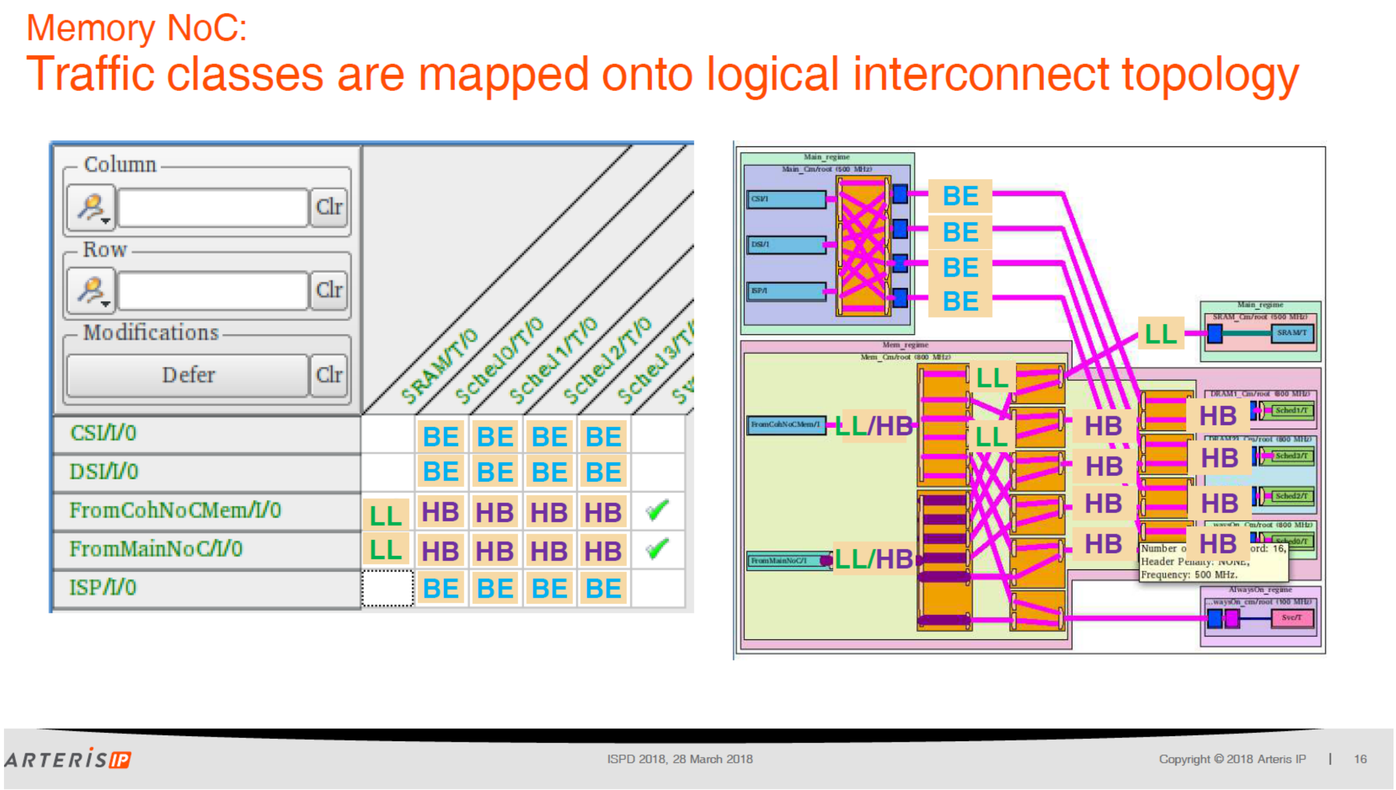
Column
Clr
Row
Clr
Modifications
Defer
Clr

	SRAM/T/0	ScheJ0/T/0	ScheJ1/T/0	ScheJ2/T/0	ScheJ3/T/	S
CSI/I/0		BE	BE	BE	BE	
DSI/I/0		BE	BE	BE	BE	
FromCohNoCMem/I/0	LL	HB	HB	HB	HB	✓
FromMainNoC/I/0	LL	HB	HB	HB	HB	✓
ISP/I/0		BE	BE	BE	BE	

ARTERIS IP
ISPD 2018, 28 March 2018
Copyright © 2018 Arteris IP | 13

U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]



Memory NoC:
Traffic classes are mapped onto logical interconnect topology

Column

Clr

Row

Clr

Modifications

Defer

Clr

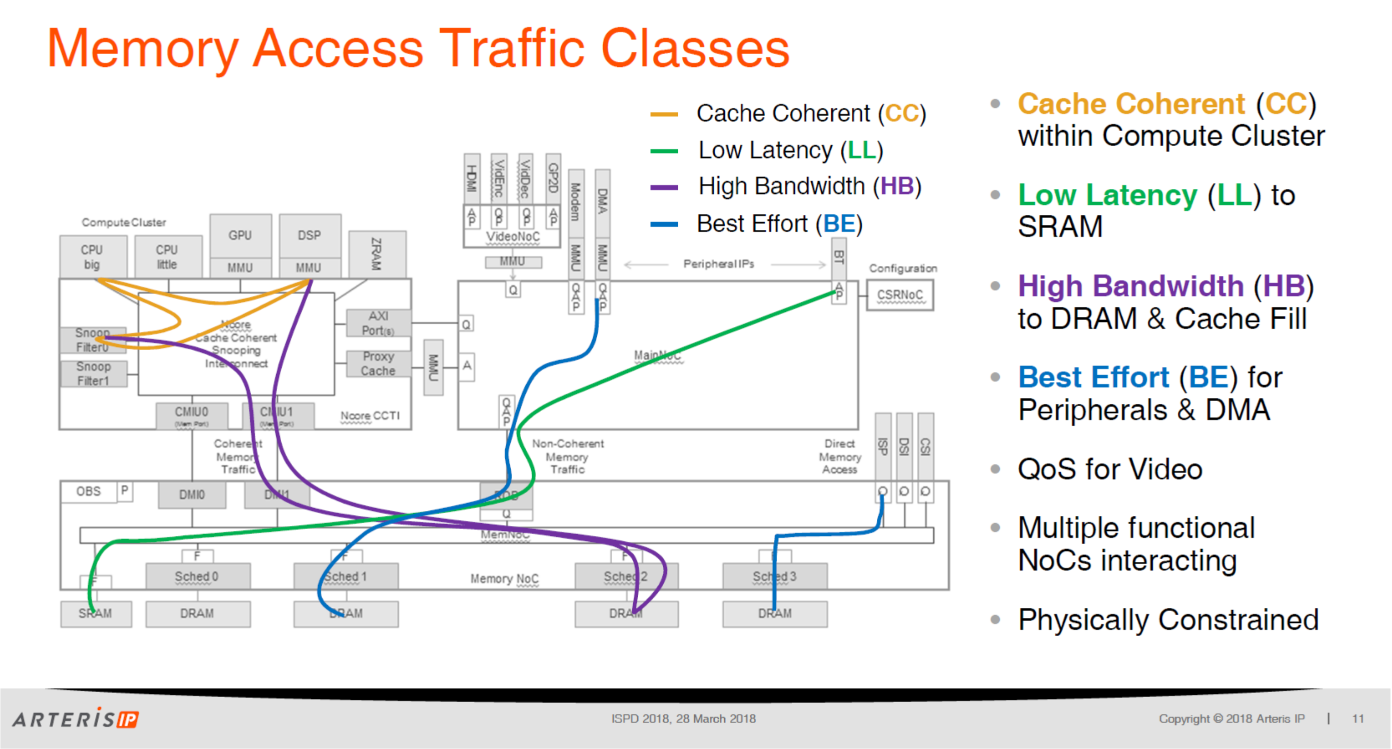
	SRAM/T/0	Sched0/T/0	Sched1/T/0	Sched2/T/0	Sched3/T/	Sv
CSI/I/0		BE	BE	BE	BE	
DSI/I/0		BE	BE	BE	BE	
FromCohNoCMem/I/0	LL	HB	HB	HB	HB	✓
FromMainNoC/I/0	LL	HB	HB	HB	HB	✓
ISP/I/0		BE	BE	BE	BE	

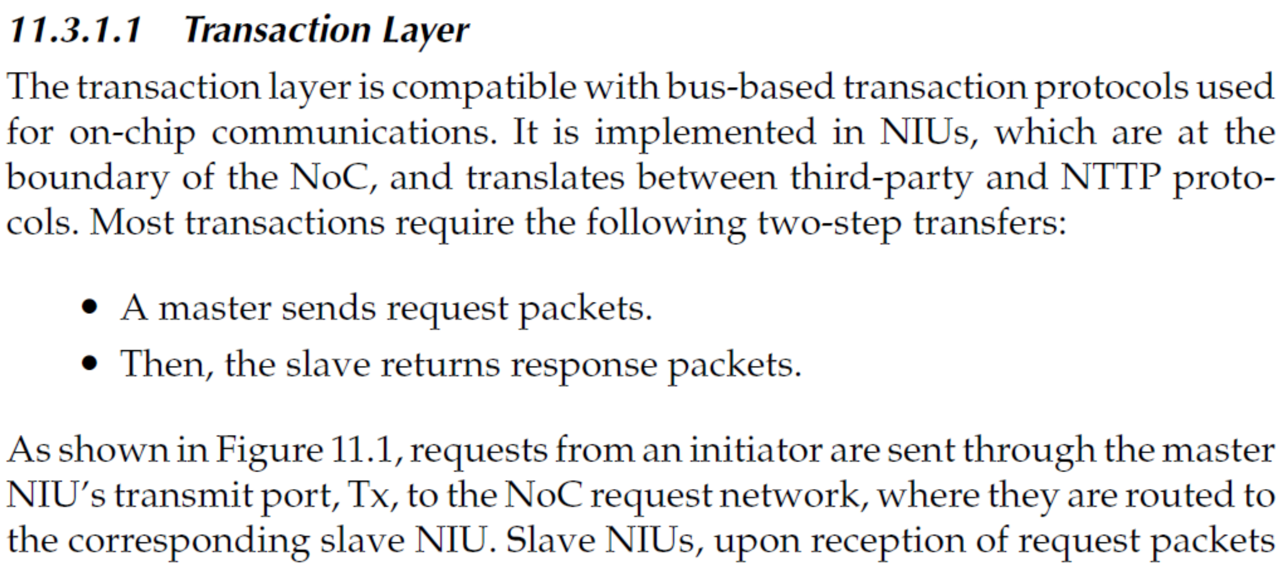
Number o... ord: 16,
Header Penalty: NONE,
Frequency: 500 MHz.

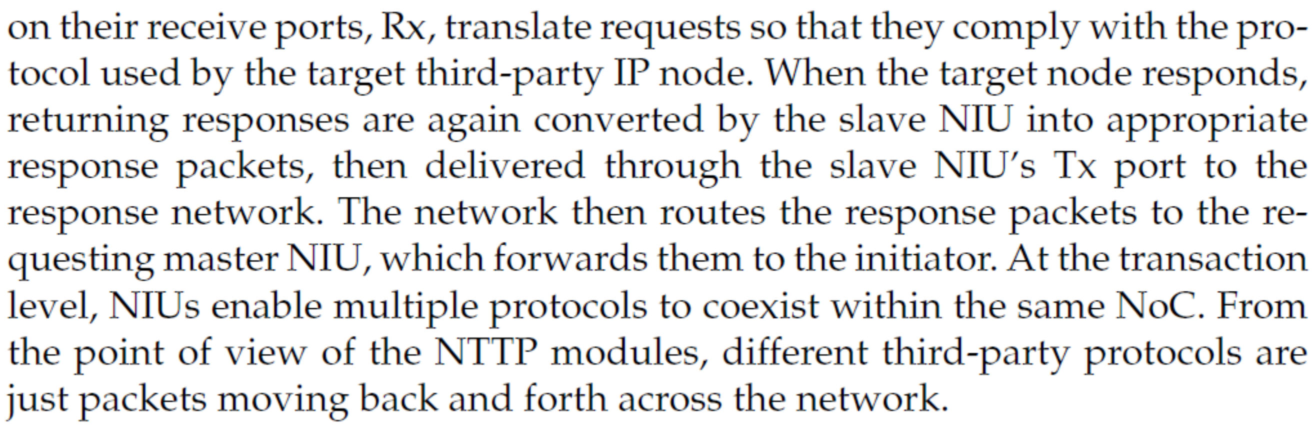
ARTERIS IP

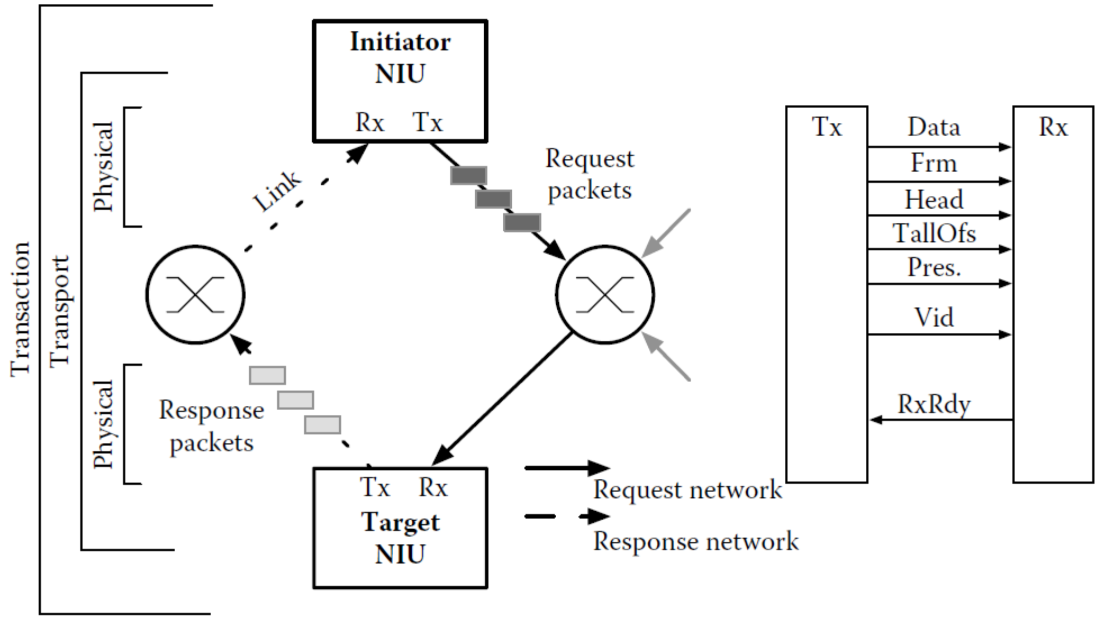
ISPD 2018, 28 March 2018

Copyright © 2018 Arteris IP | 16

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Memory Access Traffic Classes — Cache Coherent (CC) — Low Latency (LL) — High Bandwidth (HB) — Best Effort (BE) Compute Cluster CPU big CPU little GPU DSP MMU MMU ZRAM Snoop Filter0 Snoop Filter1 Ncore Cache Coherent Snooping Interconnect AXI Port(s) Proxy Cache CMIU0 CMIU1 Ncore CCTI MMU HDMI VidEnc VidDec GP2D Modem DMA VideoNoC MMU Peripheral IPs BT Configuration CSRNoC MainNoC Coherent Memory Traffic Non-Coherent Memory Traffic Direct Memory Access ISP DSI CSI OBS DMI0 DMI1 MemNoC Sched 0 Sched 1 Memory NoC Sched 2 Sched 3 SRAM DRAM DRAM DRAM DRAM • Cache Coherent (CC) within Compute Cluster • Low Latency (LL) to SRAM • High Bandwidth (HB) to DRAM & Cache Fill • Best Effort (BE) for Peripherals & DMA • QoS for Video • Multiple functional NoCs interacting • Physically Constrained ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 11 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slides 11, 13, 16.
(b) arranging, at said address translation unit, the first and the second	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product arranges, at said address translation unit, the first and the second information comprising said issued message as a single address, either literally or under the doctrine of equivalents.

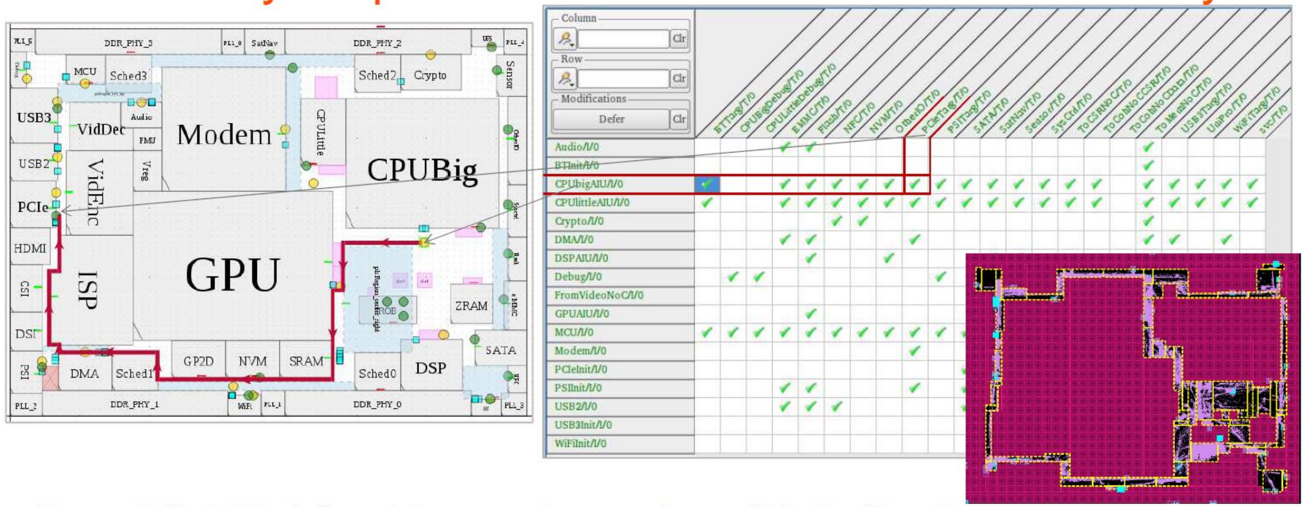
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
information comprising said issued message as a single address,	For example, the Arteris NoC used in the Snapdragon SoC included in the Motorola product uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. As a further illustration, connections between initiator module NIUs (e.g., "CPUbigAIU/1/0") and two or more target module NIUs (e.g., "ETTarg/T/0," "EMMC/T/0," "Flash/T/0," "NFC/T0," "PCIeTarg/T/0," etc.) within the Arteris NoC may be defined by a connectivity table:

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]

Connectivity Map → Interconnect Connections → Layout



	BTTag/T/0	CPUBigDebug/T/0	CPULittleDebug/T/0	EMMC/T/0	Flash/T/0	NFC/T/0	NVM/T/0	OtherIO/T/0	PCIeTag/T/0	PSITag/T/0	SATA/T/0	SatNav/T/0	Sensor/T/0	SysCtrl/T/0	To CSRNoC/T/0	To CohNoC CSR/T/0	To CohNoC Data/T/0	To MemNoC/T/0	USB3Tag/T/0	UniPro/T/0	WiFiTag/T/0	svc/T/0
Audio/I/0				✓	✓													✓				
BTInit/I/0																		✓				
CPUbigAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CPUlittleAIU/I/0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Crypto/I/0						✓	✓											✓				
DMA/I/0				✓	✓				✓									✓	✓		✓	
DSPAIU/I/0					✓			✓			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Debug/I/0		✓	✓							✓	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
FromVideoNoC/I/0											[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
GPUAIU/I/0					✓						[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MCU/I/0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Modem/I/0									✓		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
PCIeInit/I/0											✓	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
PSIInit/I/0				✓	✓				✓		✓	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
USB2/I/0				✓	✓	✓					✓	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
USB3Init/I/0											[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
WiFiInit/I/0											[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

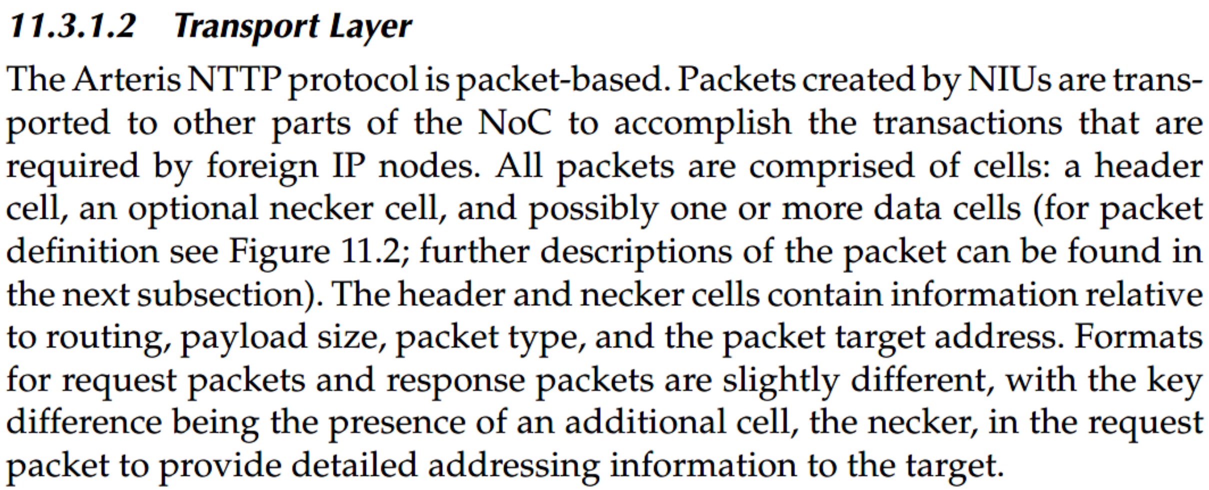
DC-Topographical

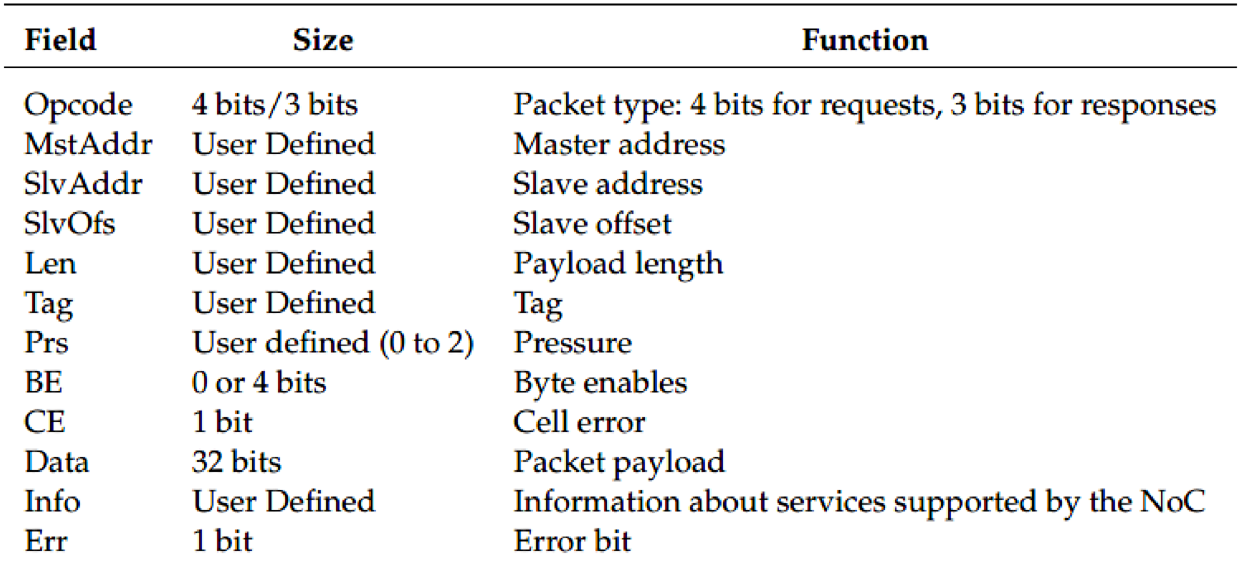
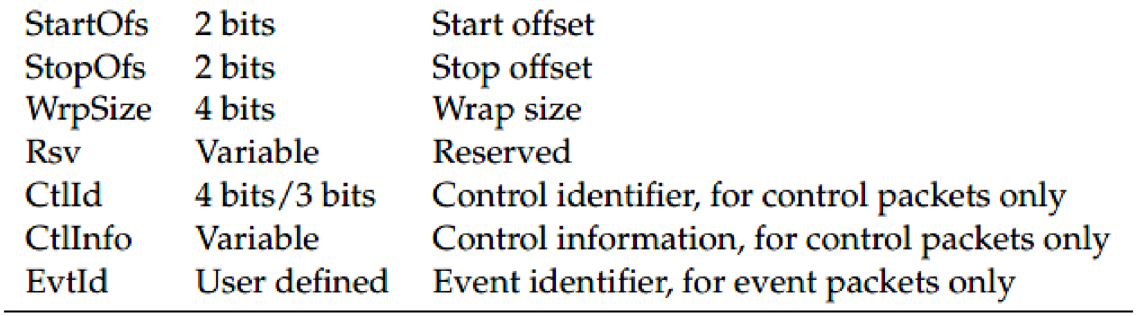
- Connectivity table defines interconnect connections within the floorplan
- Routes must pass through available channels in the floorplan
- Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU

ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 12

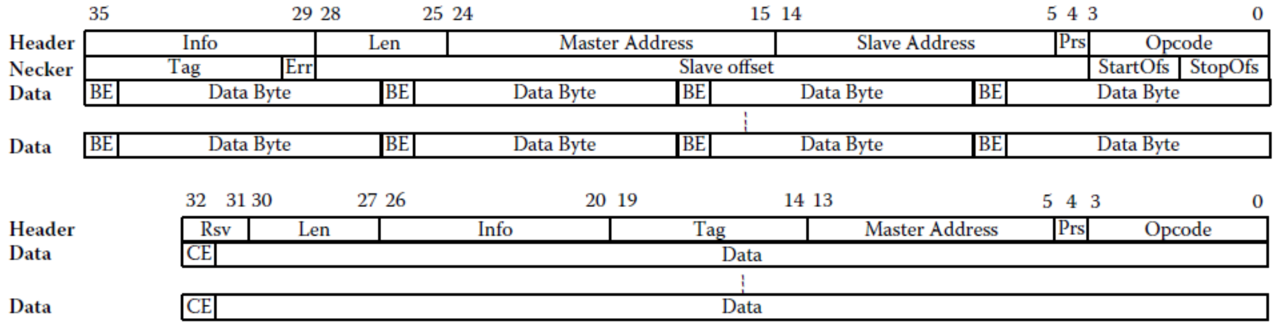
See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12.

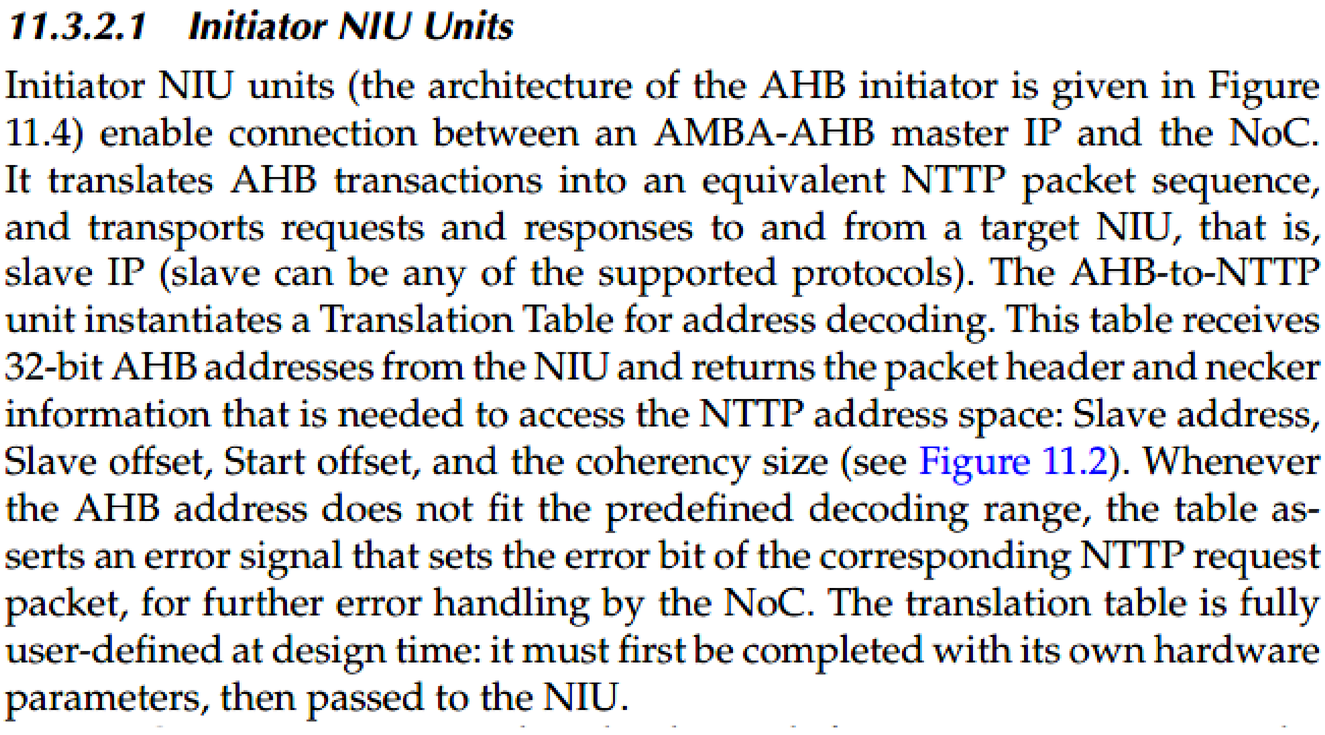
As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":

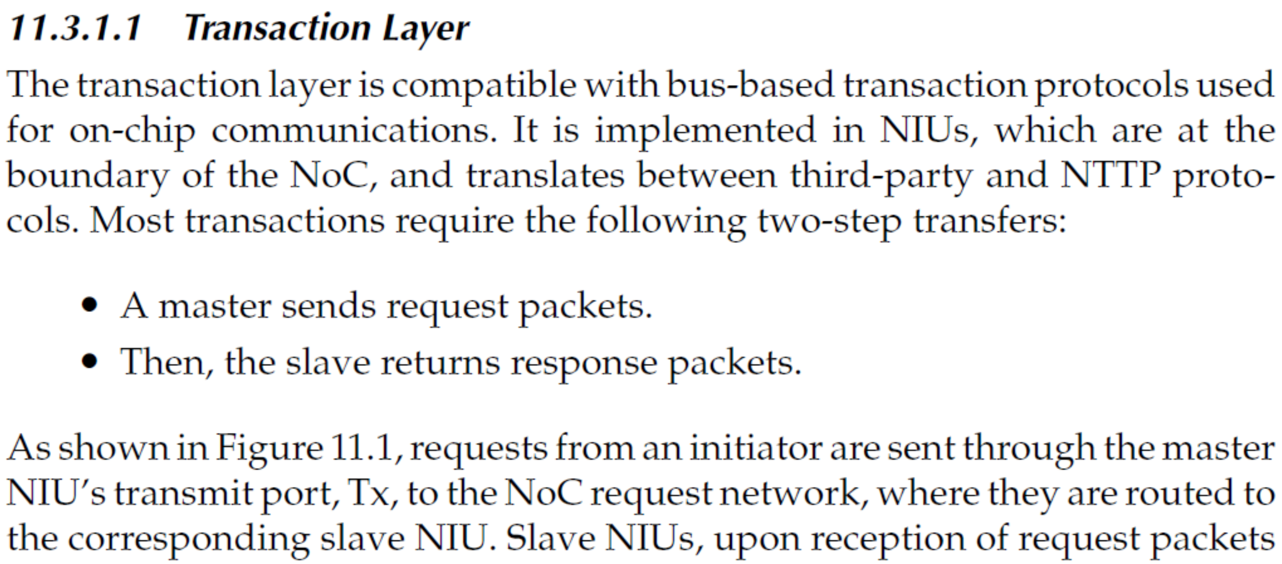
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As a further example, the packets sent in the Arteris NoC are "composed of cells that are organized into fields, with each field carrying specific information," including "Pres," "Slave address" and "Slave offset":

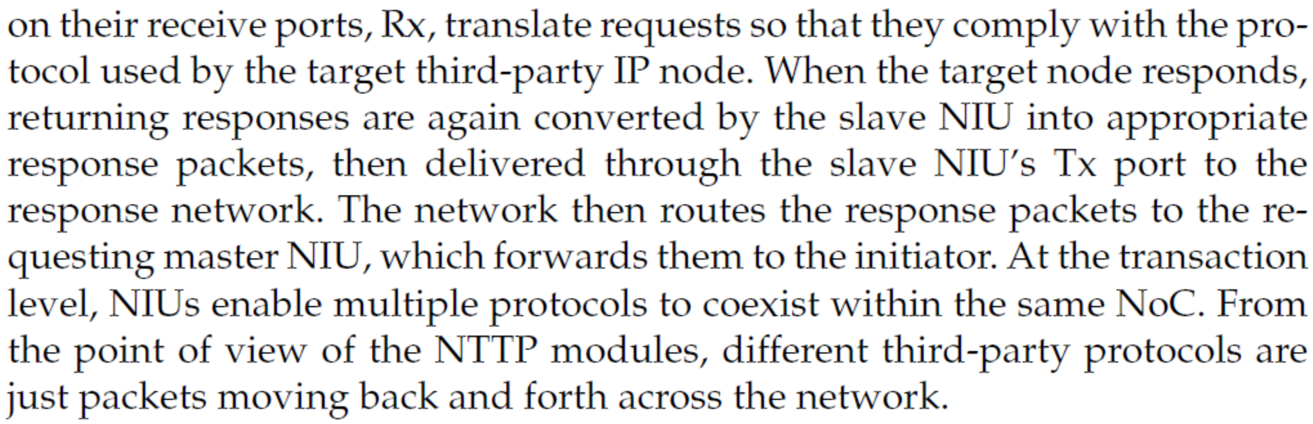
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 

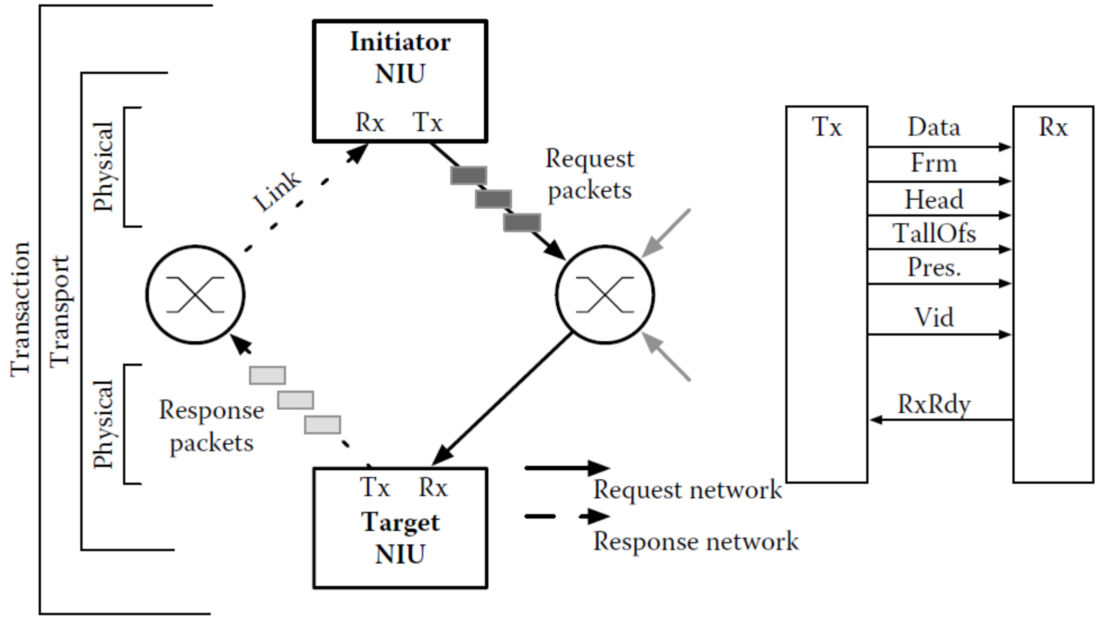
Field	Size	Function
Opcode	4 bits/3 bits	Packet type: 4 bits for requests, 3 bits for responses
MstAddr	User Defined	Master address
SlvAddr	User Defined	Slave address
SlvOfs	User Defined	Slave offset
Len	User Defined	Payload length
Tag	User Defined	Tag
Prs	User defined (0 to 2)	Pressure
BE	0 or 4 bits	Byte enables
CE	1 bit	Cell error
Data	32 bits	Packet payload
Info	User Defined	Information about services supported by the NoC
Err	1 bit	Error bit
StartOfs	2 bits	Start offset
StopOfs	2 bits	Stop offset
WrpSize	4 bits	Wrap size
Rsv	Variable	Reserved
CtlId	4 bits/3 bits	Control identifier, for control packets only
CtlInfo	Variable	Control information, for control packets only
EvtId	User defined	Event identifier, for event packets only

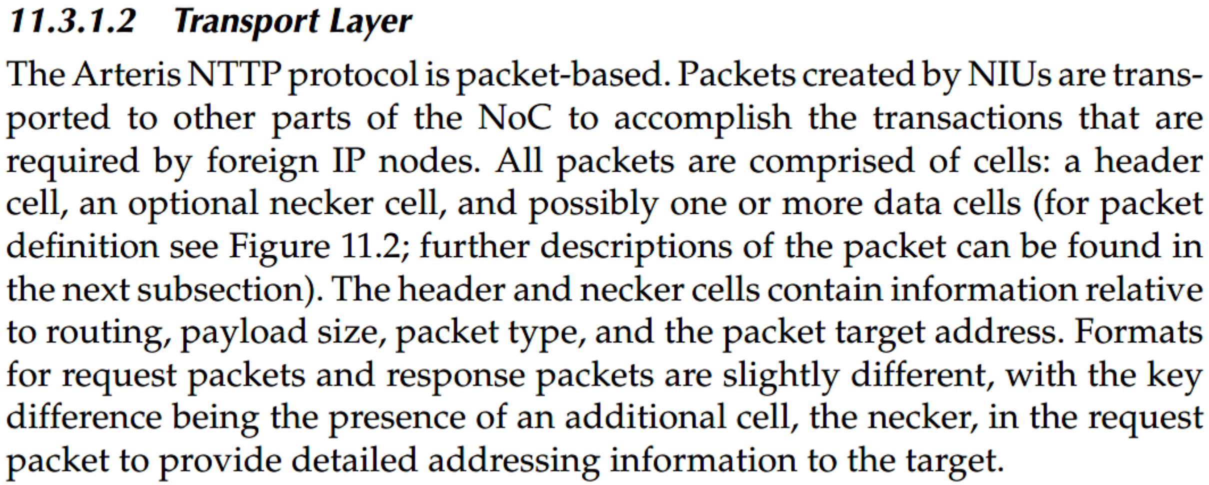
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 35 29 28 25 24 15 14 5 4 3 0 Header Info Len Master Address Slave Address Prs Opcode Necker Tag Err Slave offset StartOfs StopOfs Data BE Data Byte Data BE Data Byte 32 31 30 27 26 20 19 14 13 5 4 3 0 Header Rsv Len Info Tag Master Address Prs Opcode Data CE Data Data CE Data **FIGURE 11.2** NTTP packet structure. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 314-315. As a further example, "[i]nitiator NIU units…translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and the "AHB-to-NTTP unit instantiates a Translation Table for address decoding" with the table "receiv[ing] 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size":

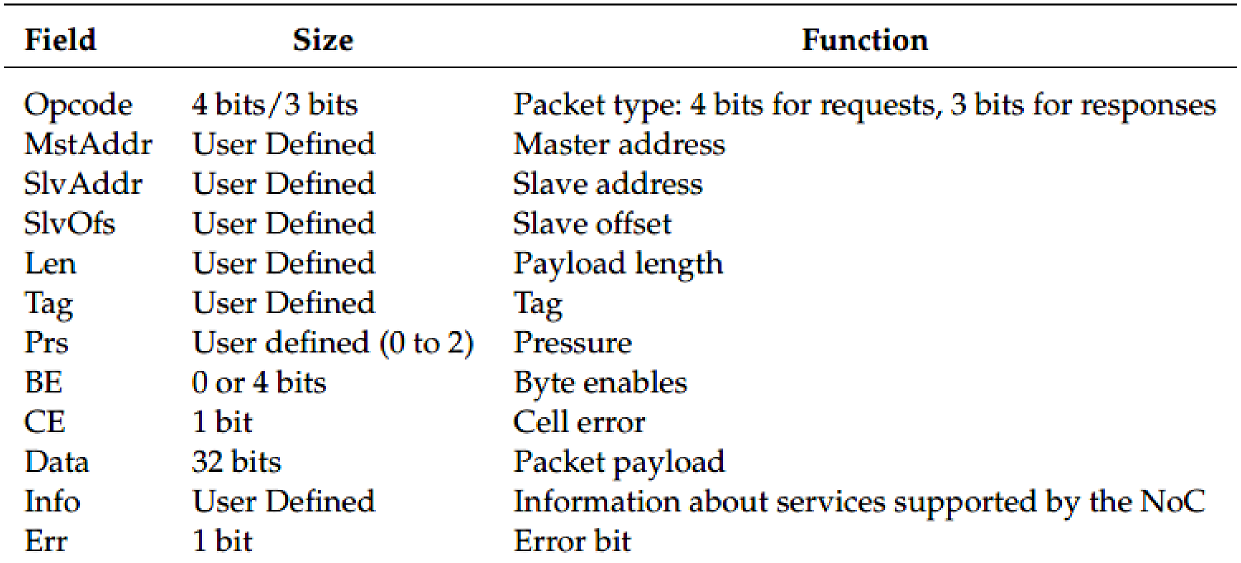
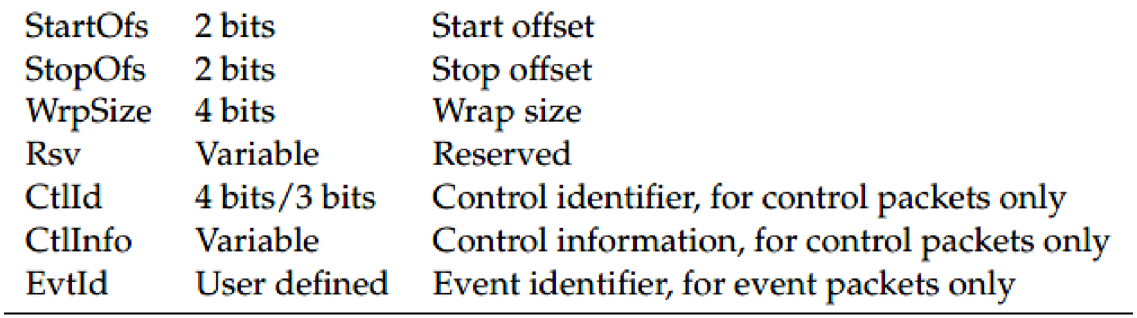
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **_11.3.2.1 Initiator NIU Units_** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317.
(c) determining, at said address translation unit, which message receiving module S is being addressed in said	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product determines, at said address translation unit, which message receiving module S is being addressed in said message request issued from said addressing module M based on said single address, either literally or under the doctrine of equivalents. For example, the Arteris NoC used by the Snapdragon SoC included in the Motorola product uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB,

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
message request issued from said addressing module M based on said single address, and	and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

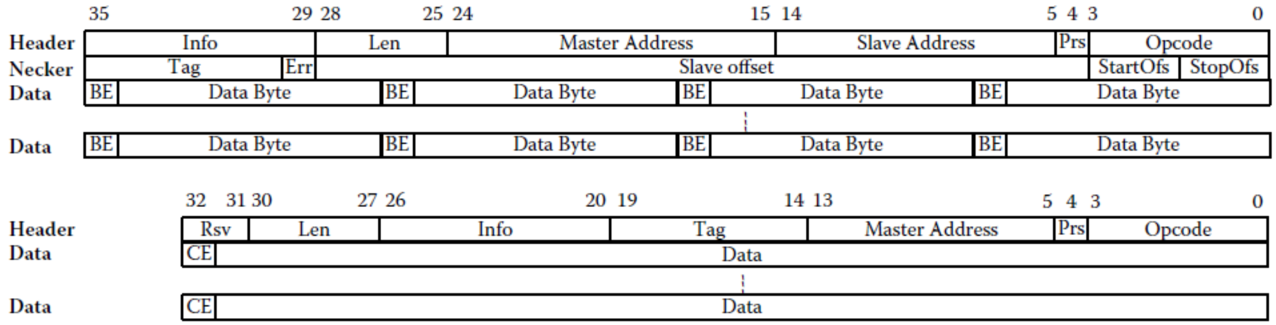
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

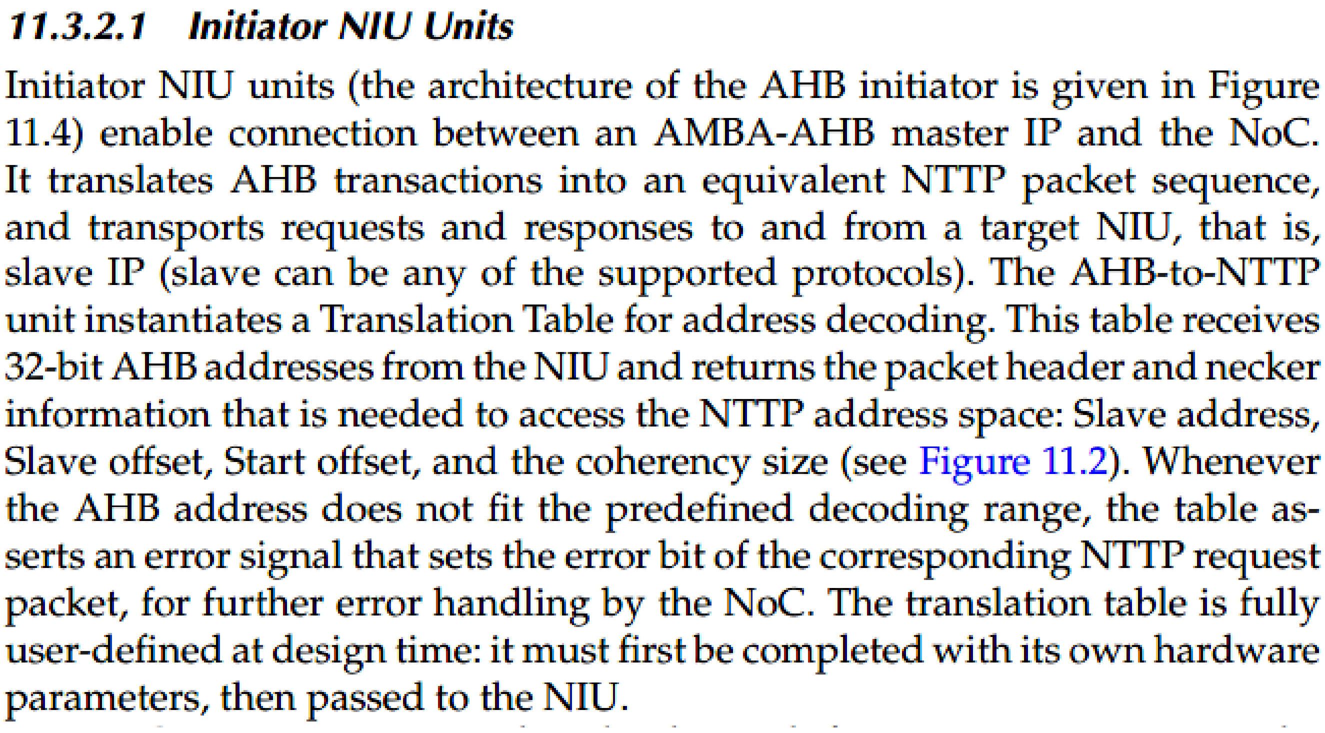
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":

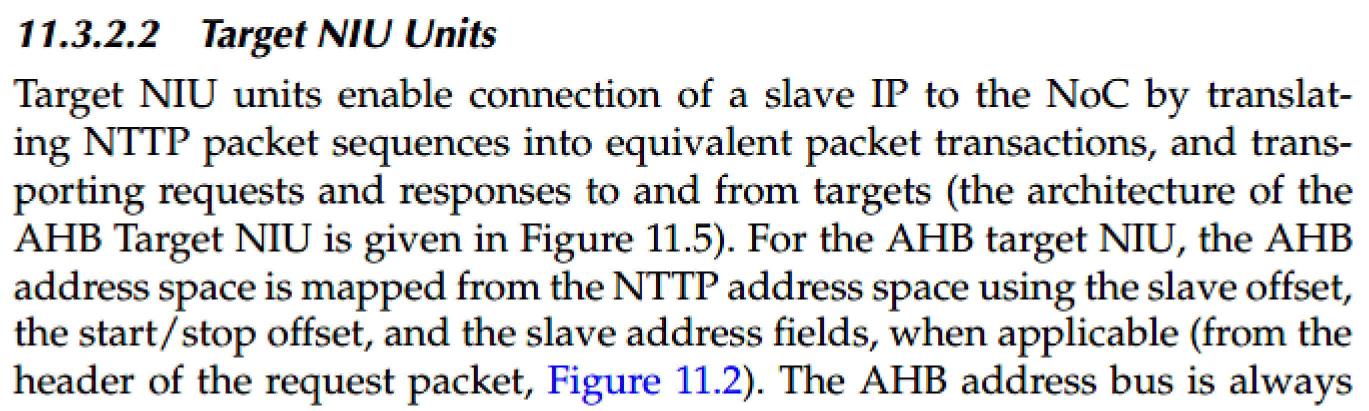
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As a further example, the packets sent in the Arteris NoC are "composed of cells that are organized into fields, with each field carrying specific information," including "Slave address" and "Slave offset":

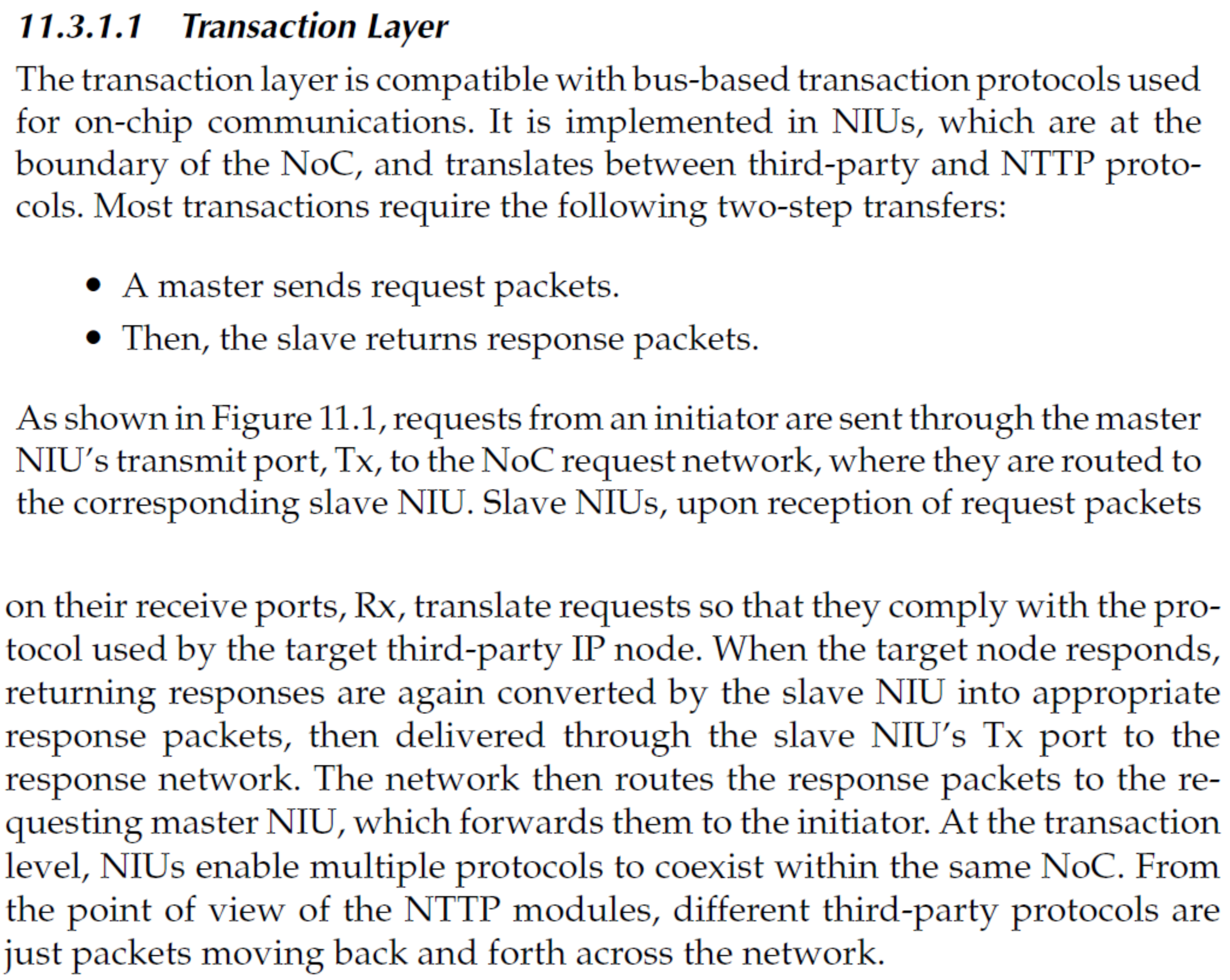
<table>
<tr><th>'9893 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip[1]</th></tr>
<tr><td></td><td>

<table>
<tr><th>Field</th><th>Size</th><th>Function</th></tr>
<tr><td>Opcode</td><td>4 bits/3 bits</td><td>Packet type: 4 bits for requests, 3 bits for responses</td></tr>
<tr><td>MstAddr</td><td>User Defined</td><td>Master address</td></tr>
<tr><td>SlvAddr</td><td>User Defined</td><td>Slave address</td></tr>
<tr><td>SlvOfs</td><td>User Defined</td><td>Slave offset</td></tr>
<tr><td>Len</td><td>User Defined</td><td>Payload length</td></tr>
<tr><td>Tag</td><td>User Defined</td><td>Tag</td></tr>
<tr><td>Prs</td><td>User defined (0 to 2)</td><td>Pressure</td></tr>
<tr><td>BE</td><td>0 or 4 bits</td><td>Byte enables</td></tr>
<tr><td>CE</td><td>1 bit</td><td>Cell error</td></tr>
<tr><td>Data</td><td>32 bits</td><td>Packet payload</td></tr>
<tr><td>Info</td><td>User Defined</td><td>Information about services supported by the NoC</td></tr>
<tr><td>Err</td><td>1 bit</td><td>Error bit</td></tr>
</table>

<table>
<tr><td>StartOfs</td><td>2 bits</td><td>Start offset</td></tr>
<tr><td>StopOfs</td><td>2 bits</td><td>Stop offset</td></tr>
<tr><td>WrpSize</td><td>4 bits</td><td>Wrap size</td></tr>
<tr><td>Rsv</td><td>Variable</td><td>Reserved</td></tr>
<tr><td>CtlId</td><td>4 bits/3 bits</td><td>Control identifier, for control packets only</td></tr>
<tr><td>CtlInfo</td><td>Variable</td><td>Control information, for control packets only</td></tr>
<tr><td>EvtId</td><td>User defined</td><td>Event identifier, for event packets only</td></tr>
</table>
</td></tr>
</table>

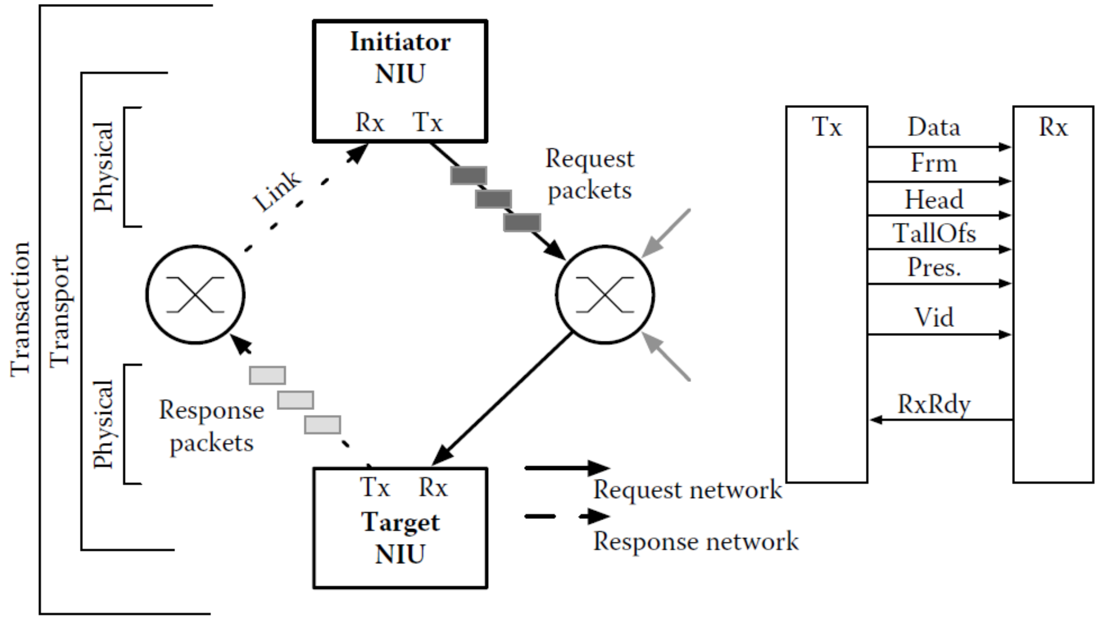
U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

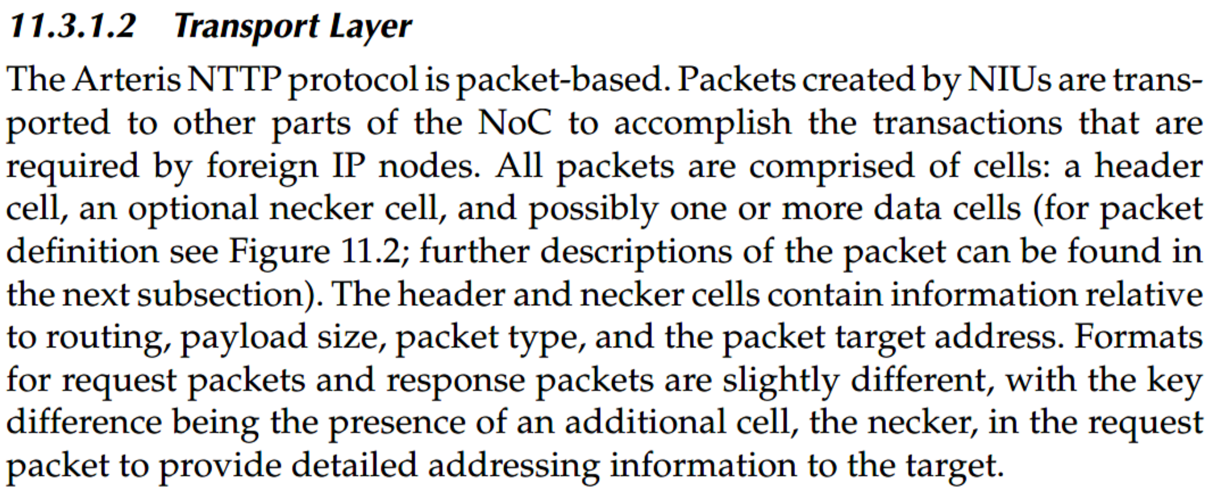
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 35 29 28 25 24 15 14 5 4 3 0 Header Info Len Master Address Slave Address Prs Opcode Necker Tag Err Slave offset StartOfs StopOfs Data BE Data Byte Data BE Data Byte 32 31 30 27 26 20 19 14 13 5 4 3 0 Header Rsv Len Info Tag Master Address Prs Opcode Data CE Data Data CE Data **FIGURE 11.2** NTTP packet structure. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 314-315. As a further example, "[i]nitiator NIU units…translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and the "AHB-to-NTTP unit instantiates a Translation Table for address decoding" with the table "receiv[ing] 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size":

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. As further example, "[f]or the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2)":

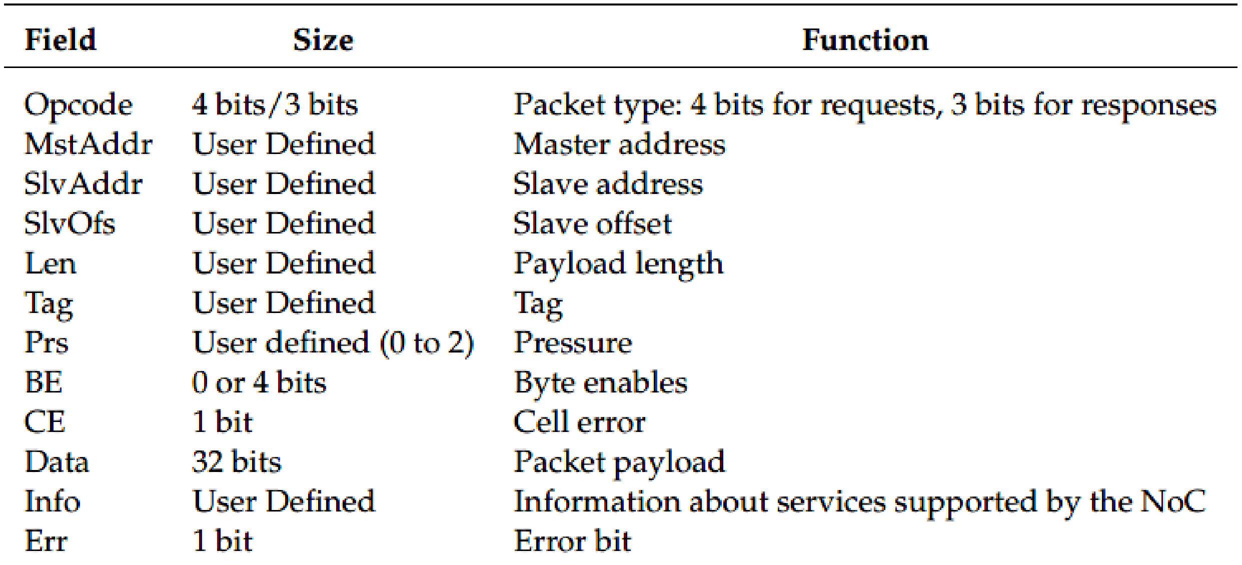
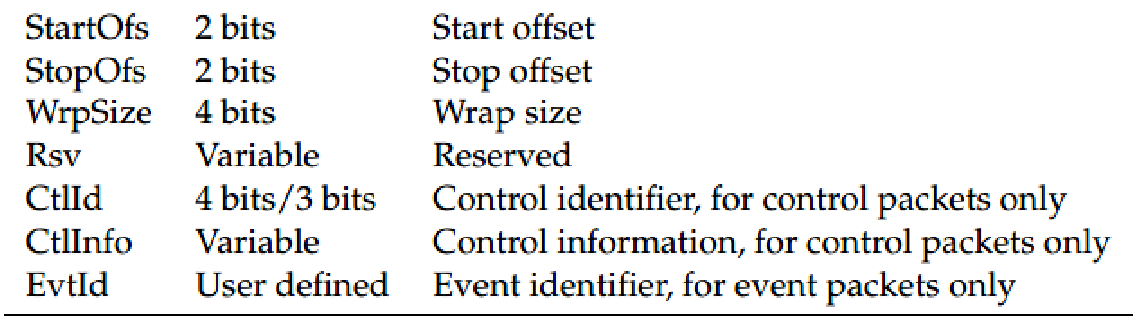
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always *Id.* at 318.
(d) further determining, at said address translation unit, the particular location within the addressed message receiving module S based on said single address.	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product further determines, at said address translation unit, the particular location within the addressed message receiving module S based on said single address, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: - A master sends request packets. - Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Initiator NIU Rx Tx Link Request packets Transaction Transport Physical Physical Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes":

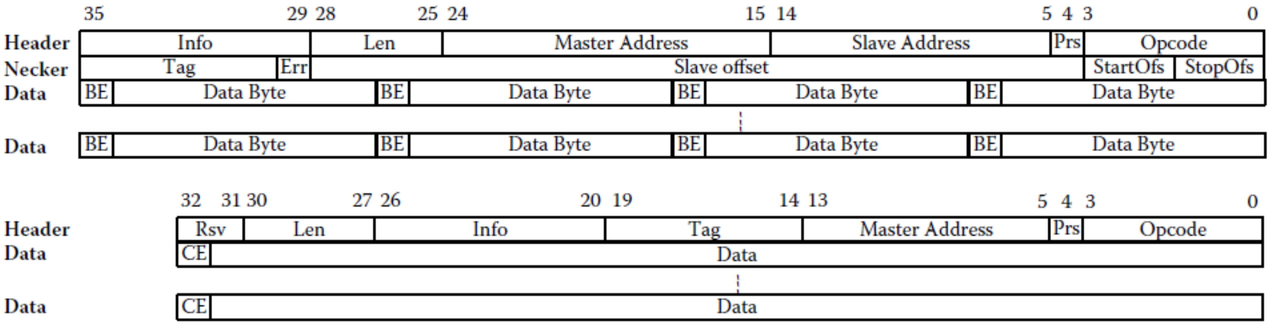
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As a further example, the packets sent in the Arteris NoC are "composed of cells that are organized into fields, with each field carrying specific information," including "Slave address" and "Slave offset":

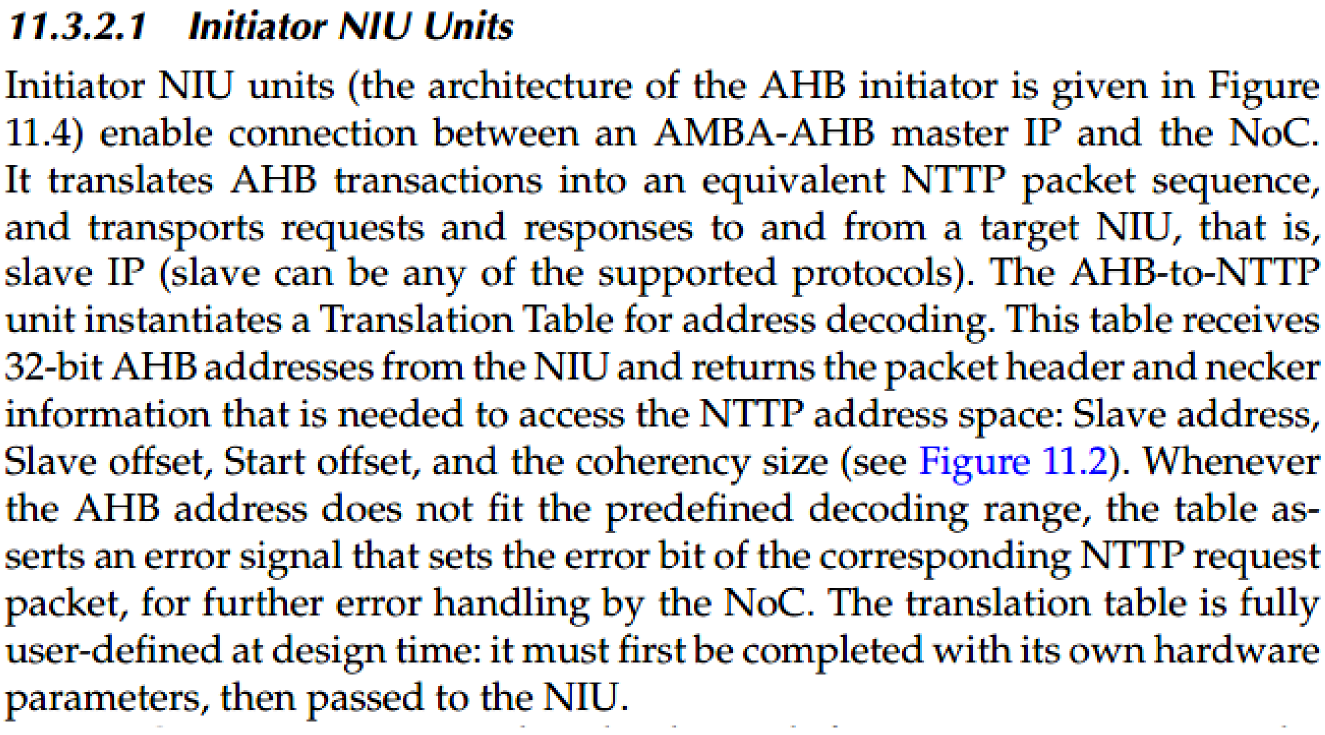
U.S. Patent No. 7,769,893 (Goossens)
"Integrated circuit and method for establishing transactions"

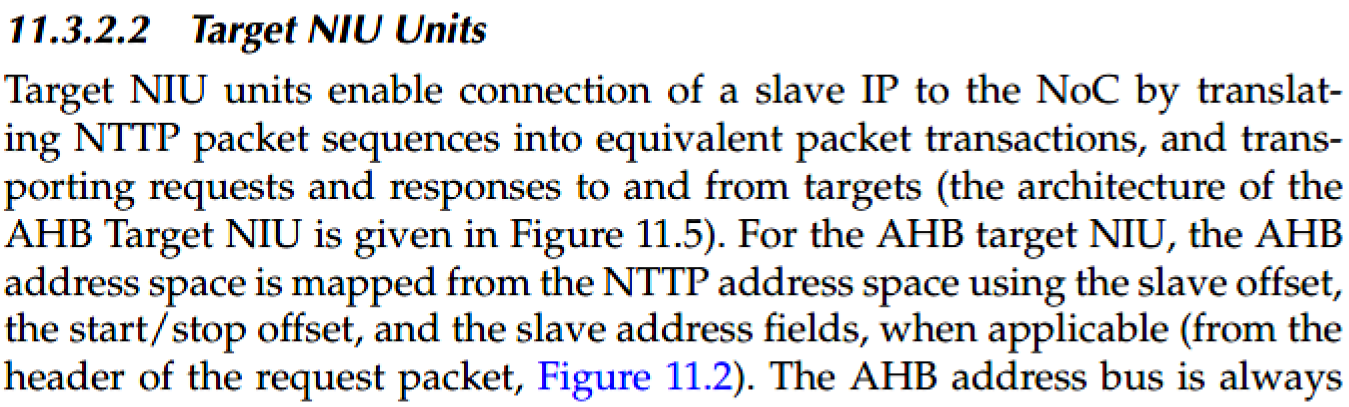
'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 

Field	Size	Function
Opcode	4 bits/3 bits	Packet type: 4 bits for requests, 3 bits for responses
MstAddr	User Defined	Master address
SlvAddr	User Defined	Slave address
SlvOfs	User Defined	Slave offset
Len	User Defined	Payload length
Tag	User Defined	Tag
Prs	User defined (0 to 2)	Pressure
BE	0 or 4 bits	Byte enables
CE	1 bit	Cell error
Data	32 bits	Packet payload
Info	User Defined	Information about services supported by the NoC
Err	1 bit	Error bit

StartOfs	2 bits	Start offset
StopOfs	2 bits	Stop offset
WrpSize	4 bits	Wrap size
Rsv	Variable	Reserved
CtlId	4 bits/3 bits	Control identifier, for control packets only
CtlInfo	Variable	Control information, for control packets only
EvtId	User defined	Event identifier, for event packets only

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 35 29 28 25 24 15 14 5 4 3 0 Header Info Len Master Address Slave Address Prs Opcode Necker Tag Err Slave offset StartOfs StopOfs Data BE Data Byte 32 31 30 27 26 20 19 14 13 5 4 3 0 Rsv CE Data **FIGURE 11.2** NTTP packet structure. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 314-315. As a further example, "[i]nitiator NIU units…translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and the "AHB-to-NTTP unit instantiates a Translation Table for address decoding" with the table "receiv[ing] 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size":

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. As further example, "[f]or the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2)":

'9893 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.2.2 Target NIU Units*** Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always *Id.* at 318.